**CS 305 Project One Template**

Document Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Comments** |
| **1.0** | **9/19/24** | **Brandon Bond** |  |

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

Brandon Bond

**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 develops individual financial plans for its customers. The plans that they provide deal with savings, retirement, investing, and insurance, this kind of information is highly sensitive and should be very secure. The value of secure communications in a finacial company is high, the information that the company has for its costumers could ruin actually ruin lives if the information is leaked. While there is no description for any international transactions, financial plans that deal in investing will suggest investments from all over the world. After a little bit of research there is not too much regarding communication restriction by the government, as long as the communication security does not threaten the United States government. With the information that is held by Artemis Financial, the main external threat now and in the future would be attackers targeting the money or identeties of the clients. Some modernized requirements to be concidered for Artemis Financial would be to do maitnence checks on the system to make sure there are not any bugs, another one would to have proper access control to the system as to properly handle access to the information.

**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.

1. Input Validation: The relevence of input validation is to verify the sensitive information with the costumer that it belongs to, this is to prevent information being given to the wrong person.
2. APIs: The relevence of APIs is to protect the data because the system is running internally and externally.
3. Cryptography: The relevence of cryptology is to encrypt the sensitive information during any transfer to keep it safe when having international communications with the financials.
4. Code Error: The relevence of error handling would help find bugs in the API and would allow for the costumer data to be more secure.
5. Code Quality: The relevence of code quality is to ensure that the code is running properly, this could include when accessed no one can run anyones information, or have access to all of the data.

**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 reviewing the code there is no input validation to be found within the customer.java, this stores the customer information and would be crucial to fix. Another issue when looking through the code is there is no error handling in the code and this could cause for unknown bugs to continue to stay in the code. There also is no encryption built into the code which could allow for users information to leak when dealing with data transfers. In the API it is not written through a POST method, this could allow for the costumer input to be exposed. While the code is written okay this doesnt mean the quality is good, missing the error handeling and input validation is an issue for the quality of the code.

**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

1. **Dependency:** bvprov-jdk15on-1.46.jar

**Vulnerability:** cpe:2.3:a:bouncycastle:legion-of-the-bouncy-castle-java-crytography-api:1.46:\*:\*:\*:\*:\*:\*:\*

**Description:** Cryptographic issues

**Solution:** update to current version

**Reference:** rest-service:compile

2. **Dependency:** spring-boot-2.2.4.RELEASE.jar

**Vulnerability:** cpe:2.3:a:vmware:spring\_boot:2.2.4:release:\*:\*:\*:\*:\*:\*

**Description:** spring-boot versions prior to version v2.2.11.RELEASE was vulnerable to temporary directory hijacking.

**Solution:** update to version 2.2.11.RELEASE or later

**Reference:** rest-service:compile

3. **Dependency:** logback-core-1.2.3.jar

**Vulnerability:** cpe:2.3:a:qos:logback:1.2.3:\*:\*:\*:\*:\*:\*:\*

**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

**Solution:** update to version 1.2.9

**Reference:** rest-service:compile

4. **Dependency:** log4j-api-2.12.1.jar

**Vulnerability:** cpe:2.3:a:apache:log4j:2.12.1:\*:\*:\*:\*:\*:\*:\*

**Description:** Improper validation of certificate with host mismatch in Apache Log4j SMTP appender.

**Solution:** update to version 2.13.2

**Reference:** rest-service:compile

5. **Dependency:** snakeyaml-1.25.jar

**Vulnerability:** cpe:2.3:a:snakeyaml\_project:snakeyaml:1.25:\*:\*:\*:\*:\*:\*:\*

**Description:** SnakeYaml's Constructor() class does not restrict types which can be instantiated during deserialization.

**Solution:** update to version 2.0 or higher

**Reference:** rest-service:runtime

6. **Dependency:** jackson-databind-2.10.2.jar

**Vulnerability:** cpe:2.3:a:fasterxml:jackson-databind:2.10.2:\*:\*:\*:\*:\*:\*:\*

**Description:** 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

**Solution:** update to 2.13.1 or later

**Reference:** rest-service:compile

7. **Dependency:** tomcat-embed-core-9.0.30.jar

**Vulnerability:** cpe:2.3:a:apache:tomcat:9.0.30:\*:\*:\*:\*:\*:\*:\*

**Description:** The refactoring present in Apache Tomcat 9.0.28 to 9.0.30, 8.5.48 to 8.5.50 and 7.0.98 to 7.0.99 introduced a regression causing invalid Transfer-Encoding headers to be incorrectly processed.

**Solution:** apply Critical Patch Update security patches as soon as possible

**Reference:** rest-service:compile

8. **Dependency:** hibernate-validator-6.0.18.Final.jar

**Vulnerability:** cpe:2.3:a:redhat:hibernate\_validator:6.0.18:\*:\*:\*:\*:\*:\*:\*

**Description:** 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.

**Solution:** apply Critical Patch Update security patches as soon as possible

**Reference:** rest-service:compile

9. **Dependency:** spring-web-5.2.3.RELEASE.jar

**Vulnerability:** cpe:2.3:a:pivotal\_software:spring\_framework:5.2.3:release:\*:\*:\*:\*:\*:\*

**Description:** Pivotal Spring Framework through 5.3.16 suffers from a potential remote code execution (RCE) issue if used for Java deserialization of untrusted data.

**Solution:** currently there is no fix for this vulnerability

**Reference:** rest-service:compile

10. **Dependency:** spring-beans-5.2.3.RELEASE.jar

**Vulnerability:** cpe:2.3:a:pivotal\_software:spring\_framework:5.2.3:release:\*:\*:\*:\*:\*:\*

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

**Solution:** Block both incoming and outgoing connections between the system and the Internet

**Reference:** rest-service:compile

11. **Dependency:** spring-webmvc-5.2.3.RELEASE.jar

**Vulnerability:** cpe:2.3:a:pivotal\_software:spring\_framework:5.2.3:release:\*:\*:\*:\*:\*:\*

**Description:** Improper Output Neutralization for Logs

**Solution:** currently there is no fix for this vulnerability

**Reference:** rest-service:compile

12. **Dependency:** spring-context-5.2.3.RELEASE.jar

**Vulnerability:** cpe:2.3:a:pivotal\_software:spring\_framework:5.2.3:release:\*:\*:\*:\*:\*:\*

**Description:** Improper Handling of Case Sensitivity

**Solution:** apply current patch

**Reference:** rest-service:compile

13. **Dependency:** spring-expression-5.2.3.RELEASE.jar

**Vulnerability:** cpe:2.3:a:pivotal\_software:spring\_framework:5.2.3:release:\*:\*:\*:\*:\*:\*

**Description:** it is possible for a user to provide a specially crafted SpEL expression that may cause a denial of service condition.

**Solution:** update to 5.2.20 or higher

**Reference:** rest-service:compile

**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.

With the missing security interventions in the code, it would be good to work on implementing and reinforcing the areas of security into the code. Overall mitigation for the vulnerabilities seem to generally be to upgrade the versions to newer versions of the dependencies to avoid risks. If no solutions are available for the vulnerabilities then patches or in code action would be suggested to mitigate any risks.