****

# Artemis Financial Vulnerability Assessment Report

Table of Contents

[Document Revision History 3](#_Toc32574607)

[Client 3](#_Toc32574608)

[Instructions 3](#_Toc32574609)

[Developer 4](#_Toc32574610)

[1. Interpreting Client Needs 4](#_Toc32574611)

[2. Areas of Security 4](#_Toc32574612)

[3. Manual Review 4](#_Toc32574613)

[4. Static Testing 4](#_Toc32574614)

[5. Mitigation Plan 4](#_Toc32574615)

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **09/18/2022** | **Tam Phan** | **Added vulnerability assessment report** |

## Client



## Instructions

Submit this completed vulnerability assessment report. Replace the bracketed text with the relevant information. In the report, identify your findings of security vulnerabilities and provide recommendations for 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 choose to include images or supporting materials. If you include them, make certain to insert them in all 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

Tam Phan

## Interpreting Client Needs

Artemis Financial expertise in financial business by developing cash flow in and out plans such as saving, retirement, investment and insurance for their customer, using their own custom software attach with RESTful API (application programing interface). To protect their client’s privacy as well as their own sensitive information externally and internally from cyber threats, they want us to find an area of security that needs to improve for future proof and identify security vulnerabilities in their code base, and create a mitigation plan to solve any issues from that point, since their most priority to focus on presently is security.

## Areas of Security

Upon checking the dependency, I found these areas are the one they need to revise based on the Vulnerability Assessment Process Flow

* Cryptography
* Input Validation
* APIs
* Code Quality
* Encapsulation

## Manual Review

Upon inspecting the pom.xl file, old Sping Boot was used; therefore, the latest patch of security will not be implemented, no input validation from the GreetingController.java which is an easy target for cross site scripting attack (XSS), the try/catch method in the DocData.java should be modified when accessing the database to avoid any SQL injection, variable account\_in customer.java should be in private with proper getter/setter method to protect customer’s information from manipulation.

## Static Testing

|  |  |  |  |
| --- | --- | --- | --- |
| Dependency Checked | Vulnerabilities | Descriptions | Attribution - Solutions |
| bcprov-jdk15on-1.46.jar | 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:legion-of-the-bouncy-castle-java-crytography-api:1.46:\*:\*:\*:\*:\*:\*:\*, cpe:2.3:a:bouncycastle:the\_bouncy\_castle\_crypto\_package\_for\_java:1.46:\*:\*:\*:\*:\*:\*:\*) : CVE-2016-1000338, CVE-2016-1000342, CVE-2016-1000343, CVE-2016-1000344, CVE-2016-1000352, CVE-2016-1000341, CVE-2016-1000345, CVE-2017-13098, CVE-2020-15522, CVE-2020-0187, CVE-2016-1000339, CVE-2020-26939, CVE-2015-7940, CVE-2018-5382, CVE-2013-1624, CVE-2016-1000346, CVE-2015-6644 | Bouncy Castle Crypto for cryptographic algorithm, problem with signature verification, DSA key pair generator, unsafe ECB mode, timing attack, padding oracle attack, private key attack | Version 1.67 and above (1.7) has no vulnerability, so it’s safe to update to that. |
| hibernate-validator-6.0.18.Final.jar | cpe:2.3:a:redhat:hibernate\_validator:6.0.18:\*:\*:\*:\*:\*:\*:\*) : CVE-2020-10693 | Hibernate’s Bean Validation. Flaws include bypass input sanitation | Fixed by updating to the latest version 7.0 |
| jackson-databind-2.10.2.jar | cpe:2.3:a:fasterxml:jackson-databind:2.10.2:\*:\*:\*:\*:\*:\*:\*, cpe:2.3:a:fasterxml:jackson-modules-java8:2.10.2:\*:\*:\*:\*:\*:\*:\*) : CVE-2020-25649, CVE-2020-36518 | General data-binding for Jackson with unsecure entity expansion which is vulnerable from XXE attacks to data integrity and DOS attack | Update to 2.13.4 |
| log4j-api-2.12.1.jar | cpe:2.3:a:apache:log4j:2.12.1:\*:\*:\*:\*:\*:\*:\*) : CVE-2020-9488 | Apache Log4j API with host mismatch can cause leak log data | The exploit has been fix in 2.12.3 and 2.13.1, so either update is suffice |
| logback-core-1.2.3.jar | cpe:2.3:a:qos:logback:1.2.3:\*:\*:\*:\*:\*:\*:\*) : CVE-2021-42550 | Logback-core module with privileges exploit | Update to 1.4 |
| snakeyaml-1.25.jar | cpe:2.3:a:snakeyaml\_project:snakeyaml:1.25:\*:\*:\*:\*:\*:\*:\*, cpe:2.3:a:yaml\_project:yaml:1.25:\*:\*:\*:\*:\*:\*:\*) : CVE-2017-18640, CVE-2022-25857, CVE-2022-38749, CVE-2022-38751, CVE-2022-38752, CVE-2022-38750 | YAML 1.1 parser and emitter with entity expansion attack, and mostly DOS attack | Update to 1.31 |
| spring-boot-2.2.4.RELEASE.jar | cpe:2.3:a:vmware:spring\_boot:2.2.4:release:\*:\*:\*:\*:\*:\*) : CVE-2022-27772 | Spring Boot vulnerable to directory hijacking | Update to 2.7.3 |
| spring-core-5.2.3.RELEASE.jar | cpe:2.3:a:springsource:spring\_framework:5.2.3:release:\*:\*:\*:\*:\*:\*, cpe:2.3:a:vmware:spring\_framework:5.2.3:release:\*:\*:\*:\*:\*:\*) : CVE-2022-22965, CVE-2021-22118, CVE-2020-5421, CVE-2022-22950, CVE-2022-22971, CVE-2022-22968, CVE-2022-22970, CVE-2021-22060, CVE-2021-22096 | Spring Core vulnerable to RCE (remote code execution) via data binding, privilege escalation, DOS attack, uppercase/lower case exploit, insertion exploit | Version below 5.3.20 is not recommended since vulnerabilities has been discovered among them |
| spring-web-5.2.3.RELEASE.jar | 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:\*:\*:\*:\*:\*:\*) : CVE-2016-1000027, CVE-2022-22965, CVE-2021-22118, CVE-2020-5421, CVE-2022-22950, CVE-2022-22971, CVE-2022-22968, CVE-2022-22970, CVE-2021-22060, CVE-2021-22096 | Spring Web share the same vulnerability with Spring Core | Update to 5.3.22 |
| tomcat-embed-core-9.0.30.jar | cpe:2.3:a:apache:tomcat:9.0.30:\*:\*:\*:\*:\*:\*:\*, cpe:2.3:a:apache\_tomcat:apache\_tomcat:9.0.30:\*:\*:\*:\*:\*:\*:\*) : CVE-2020-1938, CVE-2020-11996, CVE-2020-13934, CVE-2020-13935, CVE-2020-17527, CVE-2021-25122, CVE-2021-41079, CVE-2022-29885, CVE-2020-9484, CVE-2021-25329, CVE-2021-30640, CVE-2022-34305, CVE-2021-24122, CVE-2021-33037, CVE-2019-17569, CVE-2020-1935, CVE-2020-13943 | Core Tomcat has issue with trusted connection that led to RCE exploit, and vulnerable to high CPU usage from malicious crafted HTTP/2 requests that cause an unresponsive server and DOS | Update to 10.1.x |
| tomcat-embed-websocket-9.0.30.jar | cpe:2.3:a:apache:tomcat:9.0.30:\*:\*:\*:\*:\*:\*:\*, cpe:2.3:a:apache\_tomcat:apache\_tomcat:9.0.30:\*:\*:\*:\*:\*:\*:\*) : CVE-2020-1938, CVE-2020-8022, CVE-2020-11996, CVE-2020-13934, CVE-2020-13935, CVE-2020-17527, CVE-2021-25122, CVE-2021-41079, CVE-2022-29885, CVE-2020-9484, CVE-2021-25329, CVE-2021-30640, CVE-2022-34305, CVE-2021-24122, CVE-2021-33037, CVE-2019-17569, CVE-2020-1935, CVE-2020-13943 | Websocket Tomcat’s problems start with an incorrect default permission mostly from Linux, fail to validate TLS package that will cause DOS. It also encounters similar vulnerability to Core Tomcat | Update to 10.1.x |

## Mitigation Plan

Since every dependency has flaw so plan for this code base include updating to the latest stable version of every one of them, and make sure to implement the input validation for proper user input, implement a good encapsulation method within the code to protect sensitive data, applying a correct technique of try/catch method to prevent any SQL injections attack. Also, it’s always useful for developer to have a mindset of understanding the Vulnerability Assessment Process for proper secure coding.