****

# CS 305 Project One

**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** | **May 22, 2022** | **Byron D. Staton** |  |

## Client



## Instructions

Deliver this completed vulnerability assessment report, identifying your findings of security vulnerabilities and articulating recommendations for next steps to remedy the issues you have found.

Respond to the five steps outlined below and include your findings. Replace the bracketed text on all pages with your own words. If you choose to include images or supporting materials, be sure to insert them throughout.

## Developer

Byron Dewayne Staton

## 1. Interpreting Client Needs

Determine your client’s needs and potential threats and attacks associated with their application and software security requirements. Consider the following 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 about secure communications to consider?
* What external threats might be present now and in the immediate future?
* What are the “modernization” requirements that must be considered, such as the role of open source libraries and evolving web application technologies?

Artemis Financial assists their customers with their individual financial plans including saving, retirement, investments, and insurance. Secure communication between Artemis Financial and its patrons is of upmost importance as there is sensitive financial information being shared back and forth. At present, the company does not state any intentions on operating only in the U.S., therefore the company is likely to produce international transactions. There are no government regulations regarding secure communications, but Artemis Financial will still need to communication with all customers is secure. The company is a financial institution that will have access to sensitive information about their clients including account numbers and other financial data. In terms of modernization, Artemis Financial is strongly encouraged to keep any of the application’s open-source libraries updated to the latest version, “ensuring that security patches and bug fixes have been handled” (Selvakumar, 2020).

## 2. Areas of Security

Referring to the Vulnerability Assessment Process Flow Diagram, identify which areas of security are applicable to Artemis Financial’s software application. Justify your reasoning for why each area is relevant to the software application.

Following my review of the VAPFD, I have identified some key areas of security that are applicable to Artemis Financial’s software application.

* Input Validation: The code has areas of input in the GreetingController.java and CRUDController.java files.
* APIs: The application has RESTful API sections as indicated with REST annotations.
* Cryptography: The application will return customer account number, and balance.
* Client/Server: This code base is the server-side of the program.
* Code Error: Server prints error messages.
* Code Quality: Check for secure coding practices in the codebase.
* Dependency Check: Use latest version of spring-data-rest-webmvc and dependency-check-maven.

## 3. Manual Review

Continue working through the Vulnerability Assessment Process Flow Diagram. Identify all vulnerabilities in the code base by manually inspecting the code.

* Input Validation: GreetingController.java (line 16), CRUDController.java (line 13)
* APIs: GreetingController.java (lines 9 and 15), CRUDController (lines 7 and 12), RestServiceApplication (line 6)
* Cryptography: customer.java (lines 9 and 13)
* Client/Server
* Code Error: DocData.java (lines 25 to 30)
* Code Quality
* Dependency Check: pom.xml (line 60), spring-data-rest-webmvc (not found)

## 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 dependency check report. Include the following:

1. The names or vulnerability codes of the known vulnerabilities
2. A brief description and recommended solutions provided by the dependency check report
3. Attribution (if any) that documents how this vulnerability has been identified or documented previously

|  |  |  |
| --- | --- | --- |
| **Dependency** | **Description & Solution** | **Attribution** |
| bcprov-jdk15on-1.46.jar | The Transport Layer Security (TLS) protocol contains an issue with considering attacks to exploit secrets from a chip or system. The recommended solution is to update Bouncy Castle to v1.48. | "[Red Hat Bugzilla – Bug 908428](https://bugzilla.redhat.com/show_bug.cgi?id=908428)" by Red Hat Bugzilla |
| spring-boot-2.2.4.RELEASE.jar | The Spring OXM wrapper in versions of the framework prior to v3.2.4 and v4.0.0.M1 caused a DoS and CSRF attacks and allowed attackers to read random files. The solution is to disable default XXE processing and including an option for users to enable it if there’s a need to process XML from trusted sources. | "[CVE-2013-4152 XML External Entity (XXE) injection in Spring Framework](https://bugtraq.securityfocus.com/detail/fc84421b-04e6-4e66-9b27-714a8d434f3f)" by Red Hat Bugzilla |
| logback-core-1.2.3.jar | An attacker obtained edit configuration privileges in versions 1.2.7 and before. The attacker was able to manipulate a configuration to execute arbitrary code which loaded from LDAP servers. The solution is to upgrade to v1.2.9 or higher. | “[Possibility of vulnerability – registered as CVE-2021-42550](https://jira.qos.ch/browse/LOGBACK-1591)” by QOS.ch Jira |
| log4j-api-2.12.1.jar | There was a potential for a man-in-the-middle attack due to a host mismatch in Apache Log4j SMTP appender caused by a certificate that was not properly checked. The solution is to apply the Critical Patch Update security patches from Oracle. | “[Oracle Critical Patch Update Advisory – July 2020](https://www.oracle.com/security-alerts/cpujul2020.html)” by Oracle |
| snakeyaml-1.25.jar | The Alias feature in SnakeYAML v1.18 contained a Denial-of-Service vulnerability. The recommended solution is to upgrade to v1.26 or higher. | “[Denial of Service (DOS)](https://security.snyk.io/vuln/SNYK-JAVA-ORGYAML-537645)” by Snyk Vulnerability DB |
| jackson-databind-2.10.2.jar | Entity expansion was not secured properly in FasterXML Jackson Databind, creating a vulnerability for XXE attacks and putting the integrity of data at risk. The recommended solution is to upgrade to the latest version. | “[Red Hat Bugzilla – Bug 1887664](https://bugzilla.redhat.com/show_bug.cgi?id=1887664)” by Red Hat Bugzilla |
| tomcat-embed-core-9.0.30.jar | The internal structure of Apache Tomcat’s v9.0.28 to v9.0.30, v8.5.48 to v8.5.50 and v7.0.98 to v7.0.99 introduced a regression where its functionality returned to a previous state. As a result, there was a possibility for HTTP Request Smuggling under certain circumstances. The recommended solution is to upgrade to v7.0.100 or later. | [“[Security] CVE-2019-17569 HTTP Request Smuggling](https://lists.apache.org/thread/shxw4wz09kkq3tnbowxqo445gjhy9o4s)” by Mark Thomas |
| tomcat-embed-websocket-9.0.30.jar | The internal structure of Apache Tomcat’s v9.0.28 to v9.0.30, v8.5.48 to v8.5.50 and v7.0.98 to v7.0.99 introduced a regression where its functionality returned to a previous state. As a result, there was a possibility for HTTP Request Smuggling under certain circumstances. The recommended solution is to upgrade to v7.0.100 or later. | [“[Security] CVE-2019-17569 HTTP Request Smuggling](https://lists.apache.org/thread/shxw4wz09kkq3tnbowxqo445gjhy9o4s)” by Mark Thomas |
| hibernate-validator-6.0.18.Final.jar | A bug violated the process that creates error messages Java Bean Validation constraints where invalid EL expressions were evaluated as if they were valid. Attackers would be able to bypass input sanitation of data inputs from users or APIs. The recommended solution is to upgrade to v6.0.20.FINAL or higher. | “[In Relation To](https://in.relation.to/2020/05/07/hibernate-validator-615-6020-released/)” by Guillaume Smet |
| spring-core-5.2.3.RELEASE.jar | Pivotal Spring Framework through 5.3.16 contains a RCE issue that may or may not occur when used for recreating a Java object in memory of untrusted data. There is no solution to this vulnerability other than performing a look-ahead check for this specific case. | [“[R2] Pivotal Spring Framework HttpInvokerServiceExporter readRemoteInvocation Method Untrusted Java Deserialization](https://www.tenable.com/security/research/tra-2016-20)” by Jacob Baines |
| spring-jcl-5.2.3.RELEASE.jar | Spring Framework v5.3.0 to 5.3.16 and older unsupported versions contain a possibility for a user to craft inject a SpEL expression that will result in a Denial-of-Service vulnerability. The recommended solution is to upgrade to v5.3.17 or higher. | “[CVE-2022-22950: Spring Expression DoS Vulnerability](https://tanzu.vmware.com/security/cve-2022-22950)” by Tanzu |

## 5. Mitigation Plan

After interpreting your results from the manual review and static testing, identify the steps to remedy the identified security vulnerabilities for Artemis Financial’s software application.

* Input Validation: Validate inputs: “Ensure that user input conforms to some pattern” (Manico & Detlefsen, 2015)
* Cryptography: TLS: Use Transport Layer Security protocol to transmit sensitive data.
* Client/Server:
* Code Error: Error handling: Handle any feasible set of inputs, while enforcing proper security.
* Code Quality: Secure Coding: Develop a software architecture that enforces security policies.
* Dependency Check: OWASP Maven Dependency Check: Using v5.3.0 but can be updated to v7.1.0; Include updated version of spring-data-rest-webmvc.

**6. References**

Manico, J., & Detlefsen, A. (2015). *Iron-Clad Java: Building Secure Web Applications* (*Oracle Press*) (1st ed.) [E-book]. McGraw Hill.

Selvakumar, A. (2020, June 10). *Why upgrading software libraries is imperative*. End Point Dev. Retrieved May 22, 2022, from https://www.endpointdev.com/blog/2020/06/why-upgrading-software-libraries-is-imperative/