

# Artemis Financial Vulnerability Assessment Report

Table of Contents

[Document Revision History 3](#_Toc32574607)

[Client 3](#_Toc32574608)

[Instructions 3](#_Toc32574609)

[Developer 5](#_Toc32574610)

[1. Interpreting Client Needs 5](#_Toc32574611)

[2. Areas of Security 6](#_Toc32574612)

[3. Manual Review 8](#_Toc32574613)

[4. Static Testing 8](#_Toc32574614)

[5. Mitigation Plan 14](#_Toc32574615)

## Document Revision History

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 1.0 | 9/13/2022 | Breunna Bingham |  |

## 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

Breunna Bingham

## Interpreting Client Needs

* 1. What is the value of secure communications to the company?
  2. Does the company make any international transactions?
  3. Are there governmental restrictions about secure communications to consider?
  4. What external threats might be present now and in the immediate future?
  5. What are the modernization requirements that you must consider? For example:
     1. The role of open-source libraries
     2. Evolving web application technologies

What we know is that Artemis Financial is a financial institution dealing with several monetary transactions per client. This means the most secure communications should be utilized to ensure that their client’s information and transactions are all confidential and safe. Without having secure and reliable and secure communications, the trust from the user to the company is non-existent. While it isn’t mentioned that international transactions would take place, we can assume that it is a possibility, but for now, we will state that international transactions are not taking place at this time. There is most definitely governmental restrictions to consider when speaking about secure communications. Encryption would be the most obvious choice for protecting secure communications while information is being stored and transmitted. More specifically, it should be noted that each state has their own Data Security Laws that are in place to protect data. The Federal Trade Commission’s Act section 5 specifically, would be applicable, which requires appropriate cyber security measures and privacy laws. In addition, the ECPA (Electronic Communications Privacy Act) and SCA (Stored Communications Act) which protects the privacy of users including electronic communications, without authorization. External threats that are present now and in the immediate future are hackers. People trying to hack into the communications of the system and steal the information of users for their personal benefit or for monetary benefits. The identities of users can be stolen and sold including address information as well as SSN. This can be done in a series of ways but masking sure the code is well encapsulated and protected against unauthorized permissions and more can help to prevent these attacks from being successful. Some modernization requirements that I would consider would be an open-source library. While initially the thought of being opened sourced seemed daunting and worked against the understanding that we would need to have high security features to prevent attacks. But after some research, it seems having an open-source library has had less viral attacks and when attacks have happened, they have been eliminated very quickly due to free access and being constantly closely monitored. There are benefits to having an open-sourced library. As far as evolving web application technologies, I think it would be important to consider the ability for users to have access to the content on their phone via an app. It is more appealing to users when a company has a website that is optimized for use on a mobile device through its own application.

## Areas of Security

* Input Validation- this will ensure that all input is secure and ensure that the user is confirmed to be the user of the account. The user will have a username and password that all must be verified before accessing the account. All usernames and passwords are run through the system to be checked for requirements of specifically character counts numbers and special characters. More importantly validating all information being input can prevent against any injection attacks especially those to cause DoS from overloading the memory.
* APIs- The company already has a RESTful web API. It is important to ensure that the access that the user gains is secure. The way the API is created needs to be effective to prevent any unauthorized access into our systems to prevent attacks. Input validation can be implemented here as well.
* Cryptography- Artemis Financial sends communications through their RESTful API. We need to ensure that each session as well as all communications are encrypted to prevent any communication attacks that could leak data or user info to unauthorized individuals.
* Client/Server: Since we are using RESTful API in the system, we need to verify that all HTTP requests are verified and validated. Proper certificates need to be utilized, and verification HTTP header values.
* Code Review- All code needs to be reviewed to identify any vulnerabilities that may exist both in the input functions and the API code.
* Encapsulation- We have users accessing data within our systems all the time, because of that unauthorized access attacks are a risk. All data structures need to be protected. This adds another level of protection if an attack is successful and has access to data. The data structures can remain protected.

## Manual Review

* After reviewing the code in GreetingController.java I noticed that there is lack of proper input validation in the program. It is so important to have this to prevent injection attacks. Input validation could be accomplished through the API. There was no indication that any type of communication would be encrypted, and how. With this being a financial company encrypting such communications and data that is transmitted is imperative to the survival of the company. The RESTful API could be improved to validate HTTP requests effectively perhaps the use of POST HTTP for all communications. The use of request parameters in the GreetingController.java file and the CRUD Controller file, opens up multiple pathways for vulnerabilities.

## Static Testing

* [bcprov-jdk15on-1.46.jar](file:///C:\Users\Breezy\eclipse-workspace\rest-service\target\dependency-check-report.html#l1_991c96a4e31e6c19e2b9136c8955bd423f2dc4c7) – Bouncy Castle Crypto Package version 1.46 has a list of vulnerabilities (provided below).

Recommendation: Upgrade to Release 1.71

* + [CVE-2016-1000338](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2016-1000338)
  + [CVE-2016-1000342](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2016-1000342)
  + [CVE-2016-1000343](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2016-1000343)
  + [CVE-2016-1000344](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2016-1000344)
  + [CVE-2016-1000352](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2016-1000352)
  + [CVE-2016-1000341](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2016-1000341)
  + [CVE-2016-1000345](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2016-1000345)
  + [CVE-2017-13098](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2017-13098)
  + [CVE-2020-15522](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-15522)
  + CVE-2020-0187 (OSSINDEX)
  + [CVE-2016-1000339](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2016-1000339)
  + CVE-2020-26939 (OSSINDEX)
  + [CVE-2015-7940](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2015-7940)
  + [CVE-2018-5382](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2018-5382)
  + [CVE-2013-1624](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2013-1624)
  + [CVE-2016-1000346](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2016-1000346)
  + CVE-2015-6644 (OSSINDEX)
* [hibernate-validator-6.0.18.Final.jar](file:///C:\Users\Breezy\eclipse-workspace\rest-service\target\dependency-check-report.html#l3_7fd00bcd87e14b6ba66279282ef15efa30dd2492) – Hibernate Validator has one vulnerability listed from an improper input validator.

Recommendation: upgrade to version 6.0.20. Final

Author: Neil Griffin on 7-14-2021

* + [CVE-2020-10693](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-10693)
* [jackson-databind-2.10.2.jar](file:///C:\Users\Breezy\eclipse-workspace\rest-service\target\dependency-check-report.html#l5_0528de95f198afafbcfb0c09d2e43b6e0ea663ec) – Jackson-databind has two vulnerabilities because it lacked properly secured entity expansion which made it vulnerable to XXE attacks.

Recommendation: Upgrade to version 2.11.0

Author: Kunjan Rathod on 10-13-2020

* + [CVE-2020-25649](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-25649)
  + [CVE-2020-36518](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-36518)
* [log4j-api-2.12.1.jar](file:///C:\Users\Breezy\eclipse-workspace\rest-service\target\dependency-check-report.html#l10_a55e6d987f50a515c9260b0451b4fa217dc539cb) – Apache Log4j API has one vulnerability listed from an improper certificate validation, which could allow a man -in-the-middle attack that could leak messages being sent.

Recommendation: update to version 2.15.0-1-deb10u1

Author: Chen Zhanojun who made the discovery on 12-11-2021

* + [CVE-2020-9488](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-9488)
* [logback-core-1.2.3.jar](file:///C:\Users\Breezy\eclipse-workspace\rest-service\target\dependency-check-report.html#l12_864344400c3d4d92dfeb0a305dc87d953677c03c) – logback had only one vulnerability. Logback versions prior to 1.2.7, allow an attacker to edit configuration files allowing the execution of arbitrary code from LDAP servers.

Recommendation: update component to Logback 1.2.8

* + [CVE-2021-42550](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-42550)
* [snakeyaml-1.25.jar](file:///C:\Users\Breezy\eclipse-workspace\rest-service\target\dependency-check-report.html#l14_8b6e01ef661d8378ae6dd7b511a7f2a33fae1421) – SnakeYAML has several vulnerabilities. It allows entity expansion during a load operation.

Recommendation: Update SnakeYAML to at least version 1.26.

* + [CVE-2017-18640](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2017-18640)
  + [CVE-2022-25857](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-25857)
  + [CVE-2022-38749](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-38749)
  + [CVE-2022-38751](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-38751)
  + CVE-2022-38752 (OSSINDEX)
  + [CVE-2022-38750](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-38750)
* [spring-boot-2.2.4.RELEASE.jar](file:///C:\Users\Breezy\eclipse-workspace\rest-service\target\dependency-check-report.html#l15_225a4fd31156c254e3bb92adb42ee8c6de812714) -spring-boot versions prior to version v2.2.11. RELEASE was vulnerable to directory hijacking.

Recommendation: Update to versions v2.2.11. RELEASE or later

Author: TrungPa at unknown date

* + [CVE-2022-27772](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-27772)
* [spring-core-5.2.3.RELEASE.jar](file:///C:\Users\Breezy\eclipse-workspace\rest-service\target\dependency-check-report.html#l16_3734223040040e8c3fecd5faa3ae8a1ed6da146b) – vulnerable to remote code execution via data binding.

Recommendation: Upgrade to version 5.2.20+

* + [CVE-2022-22965](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-22965)
  + [CVE-2021-22118](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-22118)
  + [CVE-2020-5421](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-5421)
  + [CVE-2022-22950](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-22950)
  + [CVE-2022-22971](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-22971)
  + [CVE-2022-22968](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-22968)
  + [CVE-2022-22970](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-22970)
  + [CVE-2021-22060](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-22060)
  + [CVE-2021-22096](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-22096)
* [spring-web-5.2.3.RELEASE.jar](file:///C:\Users\Breezy\eclipse-workspace\rest-service\target\dependency-check-report.html#l17_dd386a02e40b915ab400a3bf9f586d2dc4c0852c) – vulnerable to privilege escalation; attacker can read/modify files

Recommendation: Upgrade to version 5.2.15

Author: Trung Pham reported on 5-25-2021

* + [CVE-2016-1000027](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2016-1000027)
  + [CVE-2022-22965](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-22965)
  + [CVE-2021-22118](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-22118)
  + [CVE-2020-5421](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-5421)
  + [CVE-2022-22950](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-22950)
  + [CVE-2022-22971](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-22971)
  + [CVE-2022-22968](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-22968)
  + [CVE-2022-22970](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-22970)
  + [CVE-2021-22060](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-22060)
  + [CVE-2021-22096](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-22096)
* [tomcat-embed-core-9.0.30.jar](file:///C:\Users\Breezy\eclipse-workspace\rest-service\target\dependency-check-report.html#l18_ad32909314fe2ba02cec036434c0addd19bcc580) – Several vulnerabilities listed. Problem with the connections not being disabled provides a vulnerability to attackers.

Recommendation: Upgrade to version 9.0.31

* + [CVE-2020-1938](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-1938)
  + [CVE-2020-11996](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-11996)
  + [CVE-2020-13934](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-13934)
  + [CVE-2020-13935](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-13935)
  + [CVE-2020-17527](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-17527)
  + [CVE-2021-25122](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-25122)
  + [CVE-2021-41079](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-41079)
  + [CVE-2022-29885](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-29885)
  + [CVE-2020-9484](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-9484)
  + [CVE-2021-25329](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-25329)
  + [CVE-2021-30640](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-30640)
  + [CVE-2022-34305](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-34305)
  + [CVE-2021-24122](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-24122)
  + [CVE-2021-33037](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-33037)
  + [CVE-2019-17569](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2019-17569)
  + [CVE-2020-1935](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-1935)
  + [CVE-2020-13943](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-13943)
* [tomcat-embed-websocket-9.0.30.jar](file:///C:\Users\Breezy\eclipse-workspace\rest-service\target\dependency-check-report.html#l20_33157f6bc5bfd03380ebb5ac476db0600a04168d) - Several vulnerabilities listed. Problem with the connections not being disabled provides a vulnerability to attackers.

Recommendation: Upgrade to Tomcat 9.0.31 (same as above)

* + [CVE-2020-1938](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-1938)
  + [CVE-2020-8022](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-8022)
  + [CVE-2020-11996](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-11996)
  + [CVE-2020-13934](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-13934)
  + [CVE-2020-13935](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-13935)
  + [CVE-2020-17527](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-17527)
  + [CVE-2021-25122](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-25122)
  + [CVE-2021-41079](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-41079)
  + [CVE-2022-29885](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-29885)
  + [CVE-2020-9484](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-9484)
  + [CVE-2021-25329](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-25329)
  + [CVE-2021-30640](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-30640)
  + [CVE-2022-34305](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-34305)
  + [CVE-2021-24122](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-24122)
  + [CVE-2021-33037](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-33037)
  + [CVE-2019-17569](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2019-17569)
  + [CVE-2020-1935](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-1935)
  + [CVE-2020-13943](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-13943)

## Mitigation Plan

* Upgrade all dependencies to all recommended versions as outlined in each dependency above.
* Implement proper Input Validation to prevent SQL injections through the RESTful API
* Encrypt data and all communications
* Incorporate the use of POST HTTP
* Fix all request parameters

**References**

*Cybersecurity and Privacy Laws Directory.* (n.d.). Retrieved September 15, 2022, from https://www.itgovernanceusa.com/federal-cybersecurity-and-privacy-laws

L., S. (2022, July 9). *Why use open-source libraries in your project*. Cleveroad Inc. - Web and App Development Company. Retrieved September 15, 2022, from https://www.cleveroad.com/blog/check-out-the-benefits-of-open-source-library-and-open-source-software/

Manico, J. (n.d.). *Iron-Clad Java*. O’Reilly Online Learning. Retrieved September 16, 2022, from https://www.oreilly.com/library/view/iron-clad-java/9780071835886/title.html

*NVD - Vulnerabilities*. (n.d.). Retrieved September 16, 2022, from https://nvd.nist.gov/vuln

OWASP Foundation. (2010, August). *OWASP Secure Coding Practices Quick Reference Guide*. https://owasp.org/www-pdf-archive/OWASP\_SCP\_Quick\_Reference\_Guide\_v1.pdf

*Secure Coding Guidelines for Java SE*. (2022, January). Oracle. Retrieved September 15, 2022, from https://www.oracle.com/java/technologies/javase/seccodeguide.html