****

# 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** | **3/24/2024** | **Dennis Renfro** |  |

## 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

Dennis Renfro

## Interpreting Client Needs

Artemis Financial is a consulting company that develops individualized financial plans for their customers. They are seeking to modernize their operations using the most current and effective software security to ensure the secure transfer of user data using their RESTful web application programming interface. Artemis conducts international transactions so the software should protect user end both internal and external inputs and client end transactions while following the data standards of the host country and international standards set by other countries. Data encryption should be used to allow only permitted users to communicate with designated parties. The software should guard against external threats to security present day and have the ability to be upgraded and improved in order to mitigate against impending future threats.

## Areas of Security

***Input validation***- All inputs should be tested against expected criteria to ensure the input data is valid. This ensures that only quality data enters the system.

***Cryptography***-Global Rain specializes in custom software design and development for entrepeneurs, businesses, and government agencies around the world. For this reason, cryptography is necessary to ensure files and documents are secure and encryption prevents data leakage and unauthorized access. Especially when dealing with international transfers where data has to comply with host and client standards and regulations.

***API interface***- A strong API is necessary to protect the API externally with the web browser as well as internally with the application programming code. A strong API also protects the third-party users of the application.

***Code Error***- Input validation and a strong API are necessary to ensure proper error handling. Proper error handling makes it easier for end users to use it correctly, strengthens the application’s resiliency against attacks which makes it more difficult for hackers to gain unauthorized access and exploit code weaknesses.

***Code Quality***- Clean high-quality code is essential for providing stability and reliability and is less likely to be vulnerable to errors and unauthorized attacks.

## Manual Review

I manually inspected the code to identify vulnerabilities in the code base. I worked through the Vulnerability Assessment Process Flow Diagram. Visual inspection of the code appeared normal. Artemis deals with international customers and I feel a strong data encryption should be in place as many transactions will be international and the data standards of the countries has to followed.

## Static Testing

**Dependency:**

***Bcprov-jdk15on-1.46.jar***

The Bouncy Castle Cryto package is a Java implementation of cryptographic algorithms. This jar contains JCE provider and lightweight API for the Bouncy Castle Cryptography APIs for JDK 1.7.

**Vulnerability:**

***CVE-201301624 & CVE-2016-10000341***

The TLS implementation in the Bouncy Castle Java library before 1.48 and C# library before 1.8 does not properly consider timing side-channel attacks on a noncompliant MAC

Recommended solution is to upgrade to 1.5 or 1.7 version

**Vulnerability:**

***CVE-2015-6644(OSSINDEX)***

This vulnerability allows a local manipulation in the code to gain access to confidential data. It reflects unknown severity but it possess a major threat.

Recommended solution: upgrade Bouncy Castle in android to 5.1.1

LMY49F and 6.0, 2016-01-01

**Vulnerability:**

***CVE-2015-7940(OSSINDEX)***

This vulnerability targets private keys and any issues in cryptography portion of the code.

Recommended solution: to upgrade libraries to 1.5 and above.

**Vulnerability:**

***CVE-2016-1000338 & CVE-2016-1000342.***

This vulnerability targets input validation as it targets injection sequence by adding extra element in the input to crash the code. The severity is high.

Recommended solution: to upgrade the cryptography-Red hat Fuse has included a package.

**Vulnerability:**

***CVE-2016-1000343***

This vulnerability targets cryptography on versions before 1.55 JCE.

Possible solution: setting the parameters in the code with the correct values

**Vulnerability:**

***CVE-2016-1000344 & CVE-2016-1000352***

This vulnerability is ranked high in severity. It targets ECB

Possible solution: use one of the other more secure cipher modes instead ( such as CBC, CTR, CFB, or OFB)

**The Apache Log4j API**

Dependency: Log4j.api.2.12.1.jar

**Vulnerability:**

***CVE-2020-9488***

Due to a newer version being available

Possible solution: upgrade to log4j2

Dependency: snakeyami-1.25.jar

This vulnerability is high. It occurs when a set of untrusted data is parsed and a denial code is the output

Possible solution: use filter, use OWASP.

Dependencies: tomcat-embed-core-9.0.30.jar

**Vulnerability:**

***CVE-2019-17569***

This vulnerability affects Tomcats Apache version 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. The severity is medium

Possible solution: to upgrade to tomcat 9.0.31

**Vulnerability:**

***CVE-2020011996***

This vulnerability has high severity and demands immediate attention. As several http commands can overload the server or the machine where the application is running causing it to crash, which will have severe impact until system is rebooted.

Possible solution: upgrade to 9.0.36

**Vulnerability:**

***CVE-2020-13934***

This vulnerability has high severity. It is similar to CVE-2020-11996. However, it is reflecting that if a number of requests are made, out of memory exceptions can have denial of service outcome. It reflects http1 and http2.

Possible solution: download YaST online\_update or zipper patch.

**Vulnerability:**

***CVE-2020-13935***

This vulnerability could trigger infinite loops with multiple requests which would ultimately result in a denial of service. The severity is high.

Possible solution: Customers using ePO 5.10:update to ePO 5.10.0 update 9.

Customers using ePO5.9.1 and earlier. Do either of the following:

Upgrade to ePO 5.10.0 update 9 to receive fixes or the Java, Tomcat, and XSS issues.

Upgrade to ePO 5.9.1 and apply 5.9.1.Hotfix EPO-919400 to receive fixes for only the Java

And Tomcat issues.

Version 9.0.31.1-deb10u2 is recommended for upgrade

**Vulnerability:**

***CVE-2020-8022***

This vulnerability reflects that the incorrect default permissions vulnerability in the packaging of tomcat. The severity is medium.

Possible solution would be to upgrade to tomcat 9.0.35 YaST online\_update or zipper patch

**Vulnerability:**

***CVE-2020-9484***

This vulnerability has high severity level. It allows attackers to file names and contents on the server and the attacker can gain access to user storage and gain control.

Possible solution: running a command to protect from any intrusion.

Dependency: spring-core-5.2.3.RELEASE.jar

**Vulnerability:**

***CVE-2020-5421***

This vulnerability may be susceptible to RFD attacks. Depending on browser selected, the vulnerability severity level can range from medium to high.

Possible solution: upgrade to and acceptable springframe work from 4.3.29 to 5.2.9

## Mitigation Plan

Address all vulnerabilities identified in the dependency-check report.

Use mitigation plan to upgrade to current versions of all vulnerabilities and most of the vulnerabilities will be mitigated.