****

# 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** | **9/15/2022** | **Steven Colley** |  |

## 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

Steven Colley

## Interpreting Client Needs

Needs examining and help solidifying the security of their web application form external threats. Run a static tester to determine possible vulnerabilities within the web application. The company Artemis Financial is a consulting company for individual financial planning. They have a strong need for secured lines of communication to their clients. This includes verifying the server/host communication, encapsulating sensitive information just as saving, retirements, investments, and insurance. Data being sent also needs to be encrypted. A need to special care with preexisting code does persist so to not create any new vulnerabilities emerge.

## Areas of Security

The DSA does not fully validate the ASN.1 encoding of the signature on verification. The ASN.1 is a code-to-code translator that acts as a point of connection from one programming language to another. The issue is caused by the Bouncy Castle JCE and is labeled as a high risk.

Input validation error that allows for invalid EL expressions to be evaluated to valid expressions. This could be used to bypass security checkpoints that would have been put into place to prevent unauthorized access. This flaw was found in the Hibernate validator version 6.2.1 and is deemed to be a medium risk.

Improper validation of certificate to the host in Apache Log4j smtp appender. This would allow a man in the middle attack that would leave user data vulnerable to be view by malicious persons. This bug is deemed to be at low risk.

## Manual Review

The main issues that the static tester discovered must deal with how the application not only validates the user, but the connection point form the server to the client. Also, while the connection has been made the ability to have secured data. All seven methods to secure a program listed within the VAPFD are needing attention to help secure how web application. Examples include the API Bouncy Castle that doesn’t properly validate the ASN.1 signature. The error of not validating EL expressions is a flaw of an API, input validation, code error, and code quality. Finally, the Apache Log4j smtp appender bug is an example of API, code not being encrypted, encapsulation, and most importantly client/server issues because the api doesn’t properly check if the client is connected to the server and only the server.

## Static Testing

[**CVE-2016-1000338**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2016-1000338)

Bouncy castle jce with a DSA not fully validating the ASN.1 which is linking code so two different languages can work together. Deemed medium risk with the recommended fix being to have a security update but couldn’t figure out what needed to be updated. I would recommend determining what characters are being entered that should be invalid and marking them as such.

[**CVE-2020-10693**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-10693)

A flaw with the hibernate validator version 6.1.2 allows for invalid EL expressions to be evaluated as valid. Allows attackers to bypass input sanitation. Within a thread of Bugzilla.redhat.com several mentions have been claimed that the issue is fixable with a download of the Red Hat Fuse 7.9.0.

[**CVE-2020-9488**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-9488)

Improper validation of certificate with host mismatch. Meaning that the client to server connection wasn’t properly validated when creating the connection. Leaving the ability for a man in the middle attack. Recommended fix is too to download latest version of Apache Log4j 2.12.3 and 2.13.1.

## Mitigation Plan

Recommendation was listed above as well but to summarize I would check and see if all security APIs are up to date and rerun the static test to ensure tests pass. Also, for any errors with no known fixes I would self-code in fixes until API software updates can be made. For instance eliminated all input validations that are allowing improper characters/sequences to be used.