****

# 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** | 2024-03-16 | **John Vowcicefski** | Initial report |



**Instruction**

This report identifies security vulnerabilities within Artemis Financial's web application and software infrastructure. Recommendations for mitigation are provided to enhance the security posture and protect against potential threats.

**Developer**

John Vowcicefski

**1. Interpreting Client Needs**

Artemis Financial requires a comprehensive security assessment to identify vulnerabilities within their web application and software infrastructure. The primary objective is to safeguard sensitive financial data against unauthorized access, data breaches, and other cyber threats. Ensuring the confidentiality, integrity, and availability of client data is paramount to maintaining trust and compliance with financial regulations.

**2. Areas of Security**

* **Input Validation Vulnerabilities**: Identified in Apache Tomcat (CVE-2023-45648, CVE-2023-46589) and Hibernate Validator (CVE-2020-10693), where improper input validation could lead to request smuggling or execution of arbitrary code.
* **Deserialization Flaws**: Found in Spring Framework (CVE-2016-1000027), posing a risk of remote code execution when processing untrusted data.
* **Logging Configuration Weaknesses**: Noted in Logback (CVE-2021-42550) and Log4j (CVE-2021-44228), where malicious configurations could lead to code execution or information leakage.
* **Denial of Service (DoS) Attacks**: Possible through crafted YAML documents in SnakeYAML (CVE-2022-1471) and the YAML project (CVE-2022-3064), consuming excessive resources.
* **Information Disclosure**: Through error messages in Apache Tomcat (CVE-2024-21733), potentially exposing sensitive information.

**3. Manual Review**

A manual review of the codebase and configurations revealed:

* **Use of Outdated Libraries**: Several components, including Hibernate Validator and Spring Framework, are outdated and contain known vulnerabilities.
* **Insecure Default Configurations**: Default settings in Apache Tomcat and Log4j do not provide adequate protection against known attack vectors.
* **Improper Error Handling**: Custom error pages and logging configurations could leak sensitive information under certain conditions.

**4. Static Testing**

Static analysis tools highlighted:

* **Input Validation Issues**: Automated tools confirmed the presence of input validation flaws that could be exploited via crafted input.
* **Insecure Deserialization**: Code paths were identified where untrusted data is deserialized without adequate security checks.
* **Configuration Flaws**: Misconfigurations in logging and data processing components that could lead to security vulnerabilities.

**5. Mitigation Plan**

To address the identified vulnerabilities, the following steps are recommended:

* **Update and Patch**: Upgrade all vulnerable components to their latest, secure versions. Specifically, update Apache Tomcat, Hibernate Validator, Spring Framework, Logback, and Log4j to versions that address the identified CVEs.
* **Input Validation**: Implement robust input validation across the application to prevent injection attacks and request smuggling.
* **Secure Configuration**: Review and adjust configurations for all components, especially logging frameworks, to prevent unauthorized access and information disclosure.
* **Error Handling**: Ensure that custom error pages do not disclose sensitive information and are configured to handle unexpected input gracefully.
* **Regular Security Audits**: Conduct periodic security assessments, including both manual reviews and automated scans, to identify and mitigate new vulnerabilities.