****

# 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** | **05/21/2023** | **Elijah Paulk** |  |

## 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

Elijah Paulk

## Interpreting Client Needs

Artemis Financial has expressed their desire to modernize their operations and enhance the security of their web-based software application. They are seeking to protect their organization from external threats and ensure the security of their customers' financial information. The potential threats and attacks associated with their application include unauthorized access, data breaches, injection attacks, cross-site scripting (XSS), cross-site request forgery (CSRF), and insecure communications.

a. What is the value of secure communications to the company? Secure communications are of great value to Artemis Financial as they handle sensitive financial information of their customers. Secure communications, such as encrypted data transmission over the network, protect against eavesdropping and unauthorized access to the data.

b. Does the company make any international transactions? As a consulting company with a global reach, Artemis Financial may engage in international transactions. This introduces additional security considerations, such as compliance with different data protection regulations and protection against attacks that target international financial transactions.

c. Are there governmental restrictions about secure communications to consider? Governmental restrictions on secure communications, such as encryption standards or data localization requirements, may need to be considered depending on the jurisdictions where Artemis Financial operates.

d. What external threats might be present now and in the immediate future? External threats that Artemis Financial may face include hackers attempting to gain unauthorized access to customer financial data, targeted phishing attacks, malware infections, DDoS attacks, and emerging threats specific to the financial sector, such as attacks targeting financial institutions.

## Areas of Security

Based on the functionality of Artemis Financial's web application, the following areas of security are relevant:

a. Authentication and Access Control:

This area involves ensuring that proper authentication mechanisms are in place to verify the identity of users accessing the application. Access control mechanisms should be implemented to restrict unauthorized access to sensitive resources and actions within the application.

b. Data Protection:

Data protection focuses on safeguarding the confidentiality, integrity, and availability of sensitive data. It includes measures such as encryption, secure storage, and secure transmission of data between the application and the users.

c. Input Validation and Output Encoding:

Input validation is crucial to prevent injection attacks, such as SQL injection and command injection, by validating and sanitizing user inputs. Output encoding helps prevent cross-site scripting (XSS) attacks by properly encoding user-generated content before displaying it.

d. Session Management:

Session management ensures that user sessions are securely established, maintained, and terminated. It involves protecting session identifiers, implementing session timeouts, and preventing session fixation and session hijacking attacks.

e. Error Handling and Logging:

Proper error handling and logging mechanisms help in identifying and diagnosing security issues, as well as providing relevant information for incident response and forensic analysis. Secure error handling and logging can also prevent the disclosure of sensitive information to attackers.

## Manual Review

During the manual review of Artemis Financial's codebase, the following vulnerabilities were identified:

Vulnerability 1: Description: In the file **UserManager.java**, the method **getUserById** does not validate user input properly, which can lead to an SQL injection vulnerability. Recommendation: Implement parameterized queries or prepared statements to sanitize and validate user input when constructing SQL queries.

Vulnerability 2: Description: In the file **PaymentController.java**, the method **processPayment** does not perform proper input validation, making it susceptible to cross-site scripting (XSS) attacks. Recommendation: Implement output encoding when displaying user-generated content to prevent XSS attacks.

## Static Testing

Using the dependency-check plug-in integrated into Maven, a static test was conducted on Artemis Financial's software application. The following vulnerabilities were identified:

**jackson-databind-2.10.2.jar**

**Description:**

General data-binding functionality for Jackson: works on core streaming API

**License:**

<http://www.apache.org/licenses/LICENSE-2.0.txt>

**log4j-api-2.12.1.jar**

**Description:**

The Apache Log4j API

**License:**

<https://www.apache.org/licenses/LICENSE-2.0.txt>

**logback-core-1.2.3.jar**

**Description:**

logback-core module

**License:**

[http://www.eclipse.org/legal/epl-v10.html, http://www.gnu.org/licenses/old-licenses/lgpl-2.1.html](http://www.eclipse.org/legal/epl-v10.html,%20http:/www.gnu.org/licenses/old-licenses/lgpl-2.1.html)

**snakeyaml-1.25.jar**

**Description:**

YAML 1.1 parser and emitter for Java

**License:**

Apache License, Version 2.0: http://www.apache.org/licenses/LICENSE-2.0.txt

**spring-boot-2.2.4.RELEASE.jar**

**Description:**

Spring Boot

**License:**

Apache License, Version 2.0: https://www.apache.org/licenses/LICENSE-2.0

**spring-boot-autoconfigure-2.2.4.RELEASE.jar**

**Description:**

Spring Boot AutoConfigure

**License:**

Apache License, Version 2.0: https://www.apache.org/licenses/LICENSE-2.0

**spring-boot-starter-web-2.2.4.RELEASE.jar**

**Description:**

Starter for building web, including RESTful, applications using Spring

MVC. Uses Tomcat as the default embedded container

**License:**

Apache License, Version 2.0: https://www.apache.org/licenses/LICENSE-2.0

**spring-core-5.2.3.RELEASE.jar**

**Description:**

Spring Core

**License:**

Apache License, Version 2.0: https://www.apache.org/licenses/LICENSE-2.0

**spring-web-5.2.3.RELEASE.jar**

**Description:**

Spring Web

**License:**

Apache License, Version 2.0: https://www.apache.org/licenses/LICENSE-2.0

**spring-webmvc-5.2.3.RELEASE.jar**

**Description:**

Spring Web MVC

**License:**

Apache License, Version 2.0: https://www.apache.org/licenses/LICENSE-2.0

**tomcat-embed-core-9.0.30.jar**

**Description:**

Core Tomcat implementation

**License:**

Apache License, Version 2.0: http://www.apache.org/licenses/LICENSE-2.0.txt

**tomcat-embed-websocket-9.0.30.jar**

**Description:**

Core Tomcat implementation

**License:**

Apache License, Version 2.0: http://www.apache.org/licenses/LICENSE-2.0.txt

## Mitigation Plan

To mitigate the identified security vulnerabilities, the following actions should be taken:

Vulnerability 1 (SQL injection):

* Implement parameterized queries or prepared statements to sanitize and validate user input when constructing SQL queries.
* Conduct a thorough code review to identify similar instances of improper input validation throughout the codebase.
* Provide developer training on secure coding practices to prevent future occurrences of SQL injection vulnerabilities.

Vulnerability 2 (XSS):

* Implement output encoding when displaying user-generated content to prevent XSS attacks.
* Perform input validation and filtering on user inputs to prevent the injection of malicious scripts.
* Apply context-specific output encoding to ensure proper protection against different types of XSS attacks.

Third-party library vulnerabilities:

* Regularly update all third-party libraries to their latest patched versions to mitigate known vulnerabilities.
* Monitor vulnerability databases and security advisories to stay informed about new vulnerabilities and updates for the libraries used.
* Consider implementing an automated vulnerability scanning tool to identify and track vulnerabilities in third-party libraries used.

Please note that the identified vulnerabilities should be addressed promptly to minimize the risk of exploitation and protect Artemis Financial's web-based software application and its users' data.