****

# 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** | **9/12/2023** | **Max Gilhespy** |  |

## 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

Max Gilhespy

## Interpreting Client Needs

Artemis Financial is a consulting company that develops individualized financial plans. The company handles sensitive customer information dealing with retirement, investments, savings, and insurance. Artemis Financial has expressed a need to modernize operations and wants to secure their own custom software with effective security measures that are up to date. They want to stop the chance of external threats and ensure secure communications. Information about international transactions is not explicitly stated but should be assumed. Governmental restrictions about secure communications should always be considered because this can affect data retention and security requirements. Artemis Financial uses a restful API which can be vulnerable to data interception when requests and responses are not structured securely. Because the technology of web applications is always evolving it is essential for the software at Artemis Financial to be able to stay up to date to remain secure. This also applies to any usage of open-source libraries as while the functionality may be optimal, they are untrusted sources, so appropriate security must be in place to prevent any threats from external technologies.

## Areas of Security

Input Validation – The RESTful API will accept user input. All input needs to be strictly validated. Successful validation means the system will be secure in preventing injection attacks.

APIs – the web service includes a RESTful API which is needed to communicate securely.

Cryptography – To protect company data and/or customers data proper cryptography should be used.

Client/Server - API access requires that proper certificates are used. This is to ensure data safety during transfer via https requests.

Code Error – The code must be fully reviewed concerning all input functions and all API access layer code.

Encapsulation - Data must be stored securely, only allowing access where authorized.

## Manual Review

In the CRUDController class, business names are sent as request parameters.

The request parameters are not validated.

There is no authentication system.

In the DocData class, the database connections have been hard-coded in.

The system is not using HTTPS.

## Static Testing

I updated the dependency check to the latest version (8.4.0)

A screenshot of a computer

Description automatically generated

Here are the results:

------------------------------------------------------------------------------------

Bouncy Castle version 1.46 has several vulnerabilities.

CVE-2013-1624

CVE-2015-6644

CVE-2015-7940

CVE-2016-1000338

CVE-2016-1000339

CVE-2016-1000341

CVE-2016-1000342

CVE-2016-1000343

CVE-2016-1000344

CVE-2016-1000345

CVE-2016-1000346

CVE-2016-1000352

CVE-2017-13098

CVE-2018-5382

CVE-2020-15522

CVE-2020-0187

CVE-2020-33201

Description: The vulnerabilities descriptions show that specific versions have vulnerabilities. The solution is to update beyond any versions listed.

Recommendation: update to at least 1.75

------------------------------------------------------------------------------------

Hibernate Validator 6.0.18 has one vulnerability.

CVE-2020-10693

Description: The vulnerabilities descriptions show that specific versions have vulnerabilities. The solution is to update beyond any versions listed.

Recommendation: update to at least 6.2.0

------------------------------------------------------------------------------------

jackson-databind-2.10.2.jar has several vulnerabilities.

CVE-2020-25649

CVE-2020-36518

CVE-2021-46877

CVE-2022-42004

CVE-2023-35116

Description: The vulnerabilities descriptions show that specific versions have vulnerabilities. The solution is to update beyond any versions listed.

Recommendation: update to at least 2.13.1

------------------------------------------------------------------------------------

Apache Log4j API 2.12.1 has one vulnerability.

CVE-2020-9488

Description: The vulnerabilities descriptions show that specific versions have vulnerabilities. The solution is to update beyond any versions listed.

Recommendation: update to at least 2.15.3

------------------------------------------------------------------------------------

Qos Logback Core 1.2.3 has one vulnerability.

CVE-2021-42550

Description: The vulnerabilities descriptions show that specific versions have vulnerabilities. The solution is to update beyond any versions listed.

Recommendation: update to at least 1.2.8

------------------------------------------------------------------------------------

------------------------------------------------------------------------------------

SnakeYAML 1.25 has several vulnerabilities.

CVE-2017-18640

CVE-2022-1471

CVE-2022-25857

CVE-2022-38749

CVE-2022-38751

CVE-2022-38752

CVE-2022-41854

CVE-2022-38750

Description: The vulnerabilities descriptions show that specific versions have vulnerabilities. The solution is to update beyond any versions listed. Many of these dependencies are vulnerable to DOS attacks.

Recommendation: update to at least 2.0

------------------------------------------------------------------------------------

Spring Boot 2.2.4 has several vulnerabilities.

CVE-2023-20873

CVE-2022-27772

CVE-2023-20883

Description: The vulnerabilities descriptions show that specific versions have vulnerabilities. The solution is to update beyond any versions listed. Many of these dependencies are vulnerable to DOS attacks.

Recommendation: update to at least 3.0.7

------------------------------------------------------------------------------------

Spring Boot Starter Web 2.2.4 has several vulnerabilities.

CVE-2023-20873

CVE-2022-27772

CVE-2023-20883

Description: The vulnerabilities descriptions show that specific versions have vulnerabilities. The solution is to update beyond any versions listed. Many of these dependencies are vulnerable to DOS attacks.

Recommendation: update to at least 3.0.7

------------------------------------------------------------------------------------

Spring Core 5.2.3 has several vulnerabilities.

CVE-2020-5421

CVE-2022-22965

CVE-2021-22118

CVE-2022-22950

CVE-2022-22971

CVE-2023-20861

CVE-2023-20863

CVE-2022-22968

CVE-2022-22970

CVE-2021-22060

CVE-2021-22096

Description: The vulnerabilities descriptions show that specific versions have vulnerabilities. The solution is to update beyond any versions listed. Many of these dependencies are vulnerable to DOS attacks.

Recommendation: update to at least 6.1.0

------------------------------------------------------------------------------------

Spring Web 5.2.3 has several vulnerabilities.

CVE-2016-1000027

CVE-2022-22965

CVE-2021-22118

CVE-2020-5421

CVE-2022-22950

CVE-2022-22971

CVE-2023-20861

CVE-2023-20863

CVE-2022-22968

CVE-2022-22970

CVE-2021-22060

CVE-2021-22096

Description: The vulnerabilities descriptions show that specific versions have vulnerabilities. The solution is to update beyond any versions listed.

Recommendation: update to at least 6.1.0

------------------------------------------------------------------------------------

Spring Webmvc 5.2.3 has several vulnerabilities.

CVE-2022-22965

CVE-2021-22118

CVE-2020-5421

CVE-2022-22950

CVE-2022-22971

CVE-2023-20861

CVE-2023-20863

CVE-2022-22968

CVE-2022-22970

CVE-2021-22060

CVE-2021-22096

Description: The vulnerabilities descriptions show that specific versions have vulnerabilities. The solution is to update beyond any versions listed.

Recommendation: update to at least 6.1.0

------------------------------------------------------------------------------------

Apache Tomcat Embed Core 9.0.30 has several vulnerabilities.

CVE-2019-17569

CVE-2020-11996

CVE-2020-13934

CVE-2020-13935

CVE-2020-13943

CVE-2020-17527

CVE-2020-1935

CVE-2020-1938

CVE-2020-9484

CVE-2021-25122

CVE-2021-41079

CVE-2022-29885

CVE-2022-42252

CVE-2021-25329

CVE-2021-30640

CVE-2022-34305

CVE-2023-41080

CVE-2021-24122

CVE-2021-33037

CVE-2023-28708

CVE-2021-43980

Description: The vulnerabilities descriptions show that specific versions have vulnerabilities. The solution is to update beyond any versions listed.

Recommendation: update to at least 11.0.1

------------------------------------------------------------------------------------

Apache Tomcat Embed Websocket 9.0.30 has several vulnerabilities.

CVE-2019-17569

CVE-2020-11996

CVE-2020-13934

CVE-2020-13935

CVE-2020-13943

CVE-2020-17527

CVE-2020-1935

CVE-2020-1938

CVE-2020-8022

CVE-2020-9484

CVE-2021-24122

CVE-2021-25122

CVE-2021-41079

CVE-2022-29885

CVE-2022-42252

CVE-2021-25329

CVE-2021-30640

CVE-2022-34305

CVE-2023-41080

CVE-2021-33037

CVE-2023-28708

CVE-2021-43980

Description: The vulnerabilities descriptions show that specific versions have vulnerabilities. The solution is to update beyond any versions listed.

Recommendation: update to at least 11.0.1

------------------------------------------------------------------------------------

## Mitigation Plan

We will use input validation to prevent DOS attacks.

We will update all dependencies as determined by the dependency check.

We will update dependencies appropriately to prevent data injection attacks.

We will implement a secure authentication scheme.

We will use a library to check for any problematic string patterns and use input validation to stop input of problematic strings.

We will switch to HTTPS protocol for all communications.

We will move request parameters to headers or body rather than URI.

We will remove hard-coded database connection credentials.