****

# 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** | **11/11/2023** | **Tyler Frey** |  |

## 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

Tyler Frey

## Interpreting Client Needs

* *Value of secure communications to the company:*

Artemis Financial is a consulting company that develops finical plans for their customers dealing with savings, retirement, investments and insurance. All of these plans include access to sensitive personal information use for each client. To ensure that their customers information is safe with the company, secure communications is the highest priority possible. Without secure communications when dealing with sensitive customer information many dangers are present such as identity theft and loss of customer faith in how the company is using their information.

* *International Transactions:*

As Artemis Financial is dealing with many different accounts and trying to modernize their operations it is safe to assume that they are also dealing with international transactions now, or will be in the future.

* *Governmental Restrictions to consider:*

There are governmental restrictions that Artemis Financial must consider, mainly some of the restrictions that are focused on user information use and storage, and how the company handles financial information. There are two restrictions that stick out more than others which are the Gramm-Leach-Bliley Act, and the European Union General Data Protection Regulation of 2016. The Gram-Leach-Bliley Acts primary concern is to make sure that financial institutions explain how they are sharing user information and also how they are protecting it (Ritchie & Jayanti, 2023). The European Union General Data Protection Regulation of 2016 deals with any company that process data, and its concern is ensuring that these companies follow their principals on protection and accountability (Wolford, 2023)

* *External Threats Present Now and In the Future:*

With Artemis Financial being a financial company that deals with sensitive customer data there are external threats now and this will only grow to be greater threats in the future. The want to steal personal information and use it for financial gain or to hurt users or the company in general will always be there. Artemis Financial needs to be aware of these threats and the forms that these threats may come in. There are many avenues that attackers could utilize to try to gain the information that Artemis Financial works with such as Denial of service attacks, ransomware, phishing, and vulnerability exploitation. With this information Artemis Financial should make application security one of its primary concerns, if this is not done than it will inevitably lead to an event that could cost the company clients, money, or even ruin the business as a whole.

* *Modernization Requirements:*

With web application technologies always advancing we must consider that there will always be new ways to exploit vulnerabilities of these advances. That being said Artemis Financial must consider the education of its employees on data security, and the importance of staying up to date with the latest exposed vulnerabilities and stay ahead of the curve when it comes to correcting these issues. Also, with using new technologies they should put secure data communications and API use at the front of the list of security concerns. This includes using static and dynamic testing of their applications at regular intervals and having dedicated security professionals on staff to be able to mitigate any events that could happen concerning their applications.

## Areas of Security

The most relevant areas of security from the Vulnerability Assessment Process Flow Diagram that Artemis Financial should be concerned with are:

* Input Validation- Making sure that any input is verified and correct
* APIs – Secure API interactions
* Cryptography – Encryption Use and Vulnerabilities
* Code Quality – Secure Coding Practices and Patterns

I chose input validation because this application will require ways to gather data. The data that is gathered needs to be validated to be sure it is an accepted type, and also to ensure that an attacker could not use this area as an point of entry by manipulating input. APIs were chosen due to the applications use of the spring framework. We must be sure we are using secure API interactions due to handling such sensitive information. Cryptography was chose because we are handling sensitive personal data and it is required by law to handle this data securely in transit and while we have possession of it, causing the need to encrypt all data. Code quality was chosen because we need to make sure that we are using security best practices when handling sensitive data and gathering input to help cut down the chances of attacks such as DOS from input or SQL injection from not parameterizing queries.

## Manual Review

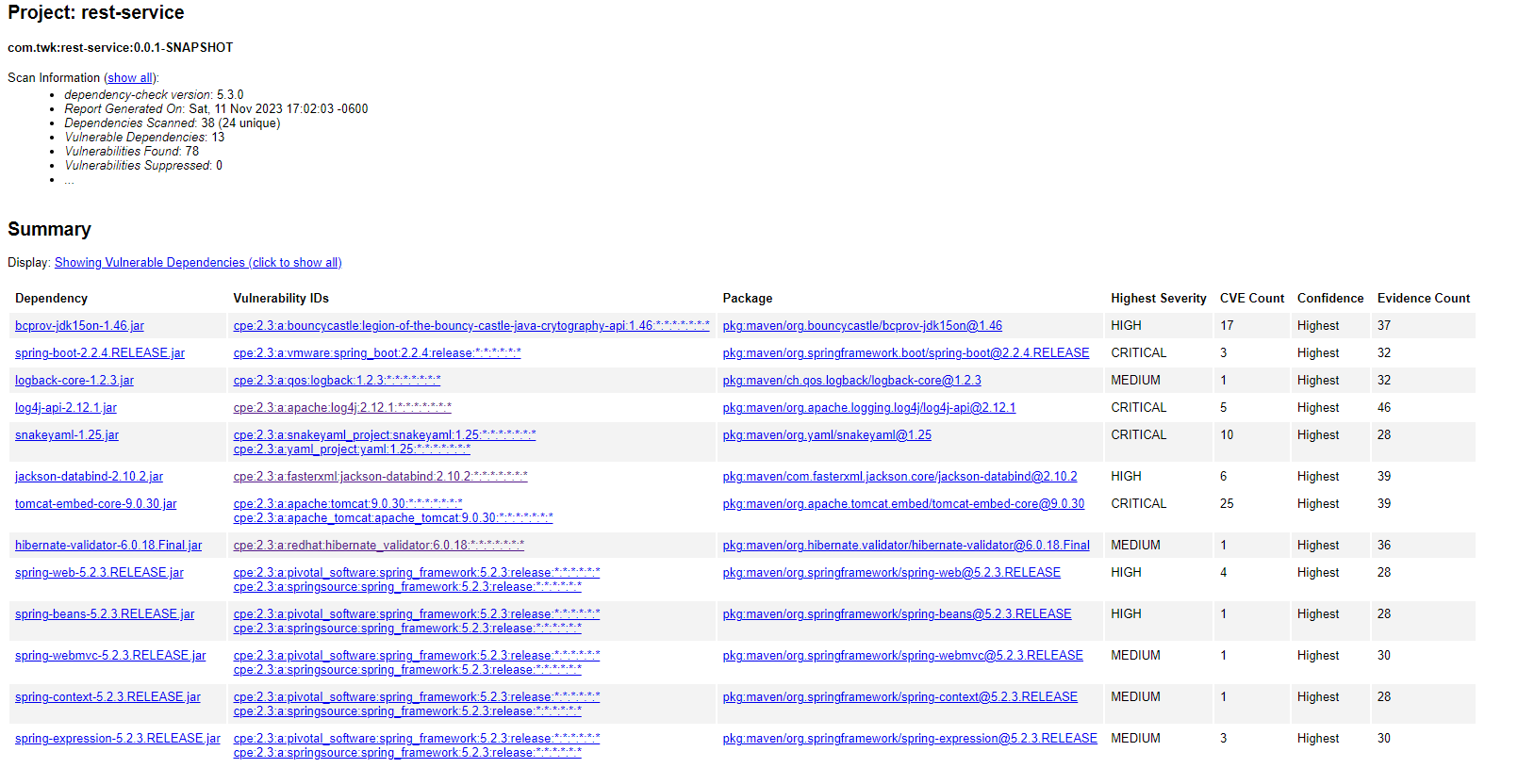
Upon manually reviewing the code bas the first thing that I noticed was that there is zero use of encryption in any of the files. The main area that made this first stick out was the CRUD.java file where we create the object to store the customers data. This is a glaring security flaw as we will be storing sensitive user data using these objects, without encryption this data would be a lot easier for an attacker to obtain.

The next flaw that I came to was the lack of input validation. The GreetingController.java file and the CRUDController.java file both take in input but have no input validation. This is another big issue as dangerous characters or strings could be passed as input allowing an attacker access to the system and all of the user’s sensitive data.

The last area I want to address is the use of the Spring framework. Using this API makes creating the application a lot easier and less time consuming. The biggest issue that I have found in the APIs use is the lack of authentication and authorization. This is something that needs to be implemented to ensure that only the allowed users can access the system, and also that the correct users with the proper authorization can access only what they are allowed to use.

## Static Testing

The following is a screenshot of the dependency-check report that was run for the Artemis Financial application:



Looking through the dependency report there were 13 vulnerable dependencies shown, with 78 total vulnerabilities found. The following is a breakdown of these vulnerabilities:

* **bcprov-jdk15on-1.46.jar – High Severity – CVE Count: 17**

Description - The Bouncy Castle Crypto package is a Java implementation of cryptographic algorithms. This jar contains JCE provider and lightweight API for the Bouncy Castle Cryptography APIs for JDK 1.5 to JDK 1.7.

CVEs - CVE-2013-1624, CVE-2015-6644 (OSSINDEX), CVE-2015-7940 (OSSINDEX), 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-2020-0187 (OSSINDEX), CVE-2020-26939 (OSSINDEX), CVE-2023-33201 (OSSINDEX)

Solution: update

* **spring-boot-2.2.4.RELEASE.jar – Critical Severity – CVE Count: 3**

Description - Spring Boot

CVEs - CVE-2022-27772, CVE-2023-20873, CVE-2023-20883

Solution: update

* **logback-core-1.2.3.jar – Medium Severity – CVE Count: 1**

Description - logback-core module

CVEs - CVE-2021-42550

Solution: update

* **log4j-api-2.12.1.jar – Critical Severity – CVE Count: 5**

Description - The Apache Log4j API

CVEs - CVE-2020-9488, CVE-2021-44228, CVE-2021-44832, CVE-2021-45046, CVE-2021-45105

Solution: update

* **snakeyaml-1.25.jar – Critical Severity – CVE Count: 10**

Description - YAML 1.1 parser and emitter for Java

CVEs - CVE-2017-18640, CVE-2021-4235, CVE-2022-1471, CVE-2022-25857, CVE-2022-3064, CVE-2022-38749, CVE-2022-38750, CVE-2022-38751, CVE-2022-38752, CVE-2022-41854

Solution: update

* **jackson-databind-2.10.2.jar – High Severity – CVE Count: 6**

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

CVEs - CVE-2020-25649, CVE-2020-36518, CVE-2021-46877, CVE-2022-42003, CVE-2022-42004, CVE-2023-35116

Solution: update

* **tomcat-embed-core-9.0.30.jar – Critical Severity – CVE Count: 25**

Description - Core Tomcat implementation

CVEs - 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-2021-25329, CVE-2021-30640, CVE-2021-33037, CVE-2021-41079, CVE-2021-43980, CVE-2022-29885, CVE-2022-34305, CVE-2022-42252, CVE-2023-28708, CVE-2023-41080, CVE-2023-42795, CVE-2023-44487, CVE-2023-45648

Solution – update

* **hibernate-validator-6.0.18.Final.jar – Medium Severity – CVE Count: 1**

Description - Hibernate's Bean Validation (JSR-380) reference implementation

CVEs - CVE-2020-10693

Solution: update

* **spring-web-5.2.3.RELEASE.jar – High Severity – CVE Count: 4**

Description - Spring Web

CVEs - CVE-2016-1000027 (OSSINDEX), CVE-2020-5421 (OSSINDEX), CVE-2021-22096 (OSSINDEX), CVE-2021-22118 (OSSINDEX)

Solution: update

* **spring-beans-5.2.3.RELEASE.jar – High Severity – CVE Count: 1**

Description - Spring Beans

CVEs - CVE-2022-22965 (OSSINDEX)

Solution: update

* **spring-webmvc-5.2.3.RELEASE.jar – Medium Severity – CVE Count: 1**

Description - Spring Web MVC

CVEs - CVE-2021-22060 (OSSINDEX)

Solution: update

* **spring-context-5.2.3.RELEASE.jar – Medium Severity – CVE Count: 1**

Description - Spring Context

CVEs - CVE-2022-22968 (OSSINDEX)

Solution: update

* **spring-expression-5.2.3.RELEASE.jar – Medium Severity – CVE Count: 3**

Description - Spring Expression Language (SpEL)

CVEs - CVE-2022-22950 (OSSINDEX), CVE-2023-20861 (OSSINDEX), CVE-2023-20863 (OSSINDEX)

Solution: update

## Mitigation Plan

These are the areas of focus for my mitigation plan for Artemis Financials web application:

* + Update all frameworks and dependencies
  + Implement input validation in all methods that take any form of input
  + Implement user authentication and authorization
  + Implement Query Parameterization to ensure database security
  + Ensure use of HTTPS protocol to secure connections and transmission of data

**References**

Ritchie, J. N. & A., & Jayanti, S. F.-T. and A. (2023, June 16). *Gramm-Leach-Bliley Act*. Federal Trade Commission. https://www.ftc.gov/business-guidance/privacy-security/gramm-leach-bliley-act

Wolford, B. (2023, September 14). *What is GDPR, the EU’s new Data Protection Law?* GDPR.eu. https://gdpr.eu/what-is-gdpr/