****

# Artemis Financial Vulnerability Assessment Report

Table of Contents

[Artemis Financial Vulnerability Assessment Report 1](#_Toc130130090)

[Document Revision History 3](#_Toc130130091)

[Client 3](#_Toc130130092)

[Developer 4](#_Toc130130093)

[1. Interpreting Client Needs 4](#_Toc130130094)

[2. Areas of Security 4](#_Toc130130095)

[3. Manual Review 5](#_Toc130130096)

[4. Static Testing 5](#_Toc130130097)

[5. Mitigation Plan 8](#_Toc130130098)

[6. Action List 9](#_Toc130130099)

[7. References 10](#_Toc130130100)

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **03/18/2023** | **David Novosad** | **Initial documentation release** |

## Client



## Developer

David Novosad

## Interpreting Client Needs

For this project, I, as a developer for Global Rain, which is a software engineering company, was assigned to work with Artemis Financial. It is a consulting company that develops individualized financial plans for its customers. These financial plans include savings, retirement, investments, and insurance. Their primary goal is to modernize their operations while keeping the security of their custom RESTful web application programming interface (API) a priority. They need to use the most current and effective software security. It will be a priority for us to develop software that is secure, as the customers’ sensitive information and money are at stake. There are possibilities that the transactions will be done internationally, which provides a higher risk as the laws and rules for software security differ between, for example, the United States, Europe, or international security laws. Depending on the client’s location, there might be local, state, and governmental restrictions in place. Each city, state, and country has its own laws, restrictions, and regulations, which would be essential to investigate before confirming the security that will have to take place in their software to satisfy those needs. Since we are working in the financial industry, it is very important to develop a secure system, as it is one of the top industries for attacks and exploitation of vulnerabilities. There might be attacks and exploits to get hold of customers’ funds or steal their sensitive data, such as their SSN, address, or contact information. These attacks could include denial of service (DoS), phishing attacks, or harmful breaches of information. The usage of open-source libraries is a risk as anyone has access to the source code, which makes it easier for hackers to explore the vulnerabilities. With the evolving web application technologies, the maintenance of the software will have to update those to make sure we are on top of any vulnerabilities that get found with the older versions.

## Areas of Security

After reviewing the Vulnerability Assessment Process Flow Diagram, I identified multiple areas of security that apply to Artemis Financial’s web application. The areas we should focus on are secure API interactions, secure input and representation, secure error handling, and secure data structures.

Making sure that the application program interface (API) is secure is an important feature. Securing the connection between the server and the users’ communication is one part of it. The other part is confirming that it is trustworthy, encrypted, and secured (this can be achieved either by the HTTP protocols or custom code for validation). Another step to improve security is input validation. Making sure the input is validated before being stored can help us ensure no malicious input gets stored or executed inside of our software. This validation can be done in the form of text stripping, a maximum input length allowance, or a ban on special characters. Correct error handling helps us respond to the issues that happen during the execution of the RESTful API calls or the execution of the code. It informs the user of the software that something went wrong. This could mean that the connection is not secure, that the system cannot encrypt the transaction, or that the input is illegal. These are just a few of the many errors and exceptions that could happen. This is important as the hacker will try to use the errors to their advantage, so the correct handling of the errors is a crucial security area. The last step is making sure that the data structures are secure. This is also referred to as encapsulation. This allows us to make sure that the code is not being updated or changed in a way that would allow the attacker to exploit the system. Making sure that the variables are set correctly, the methods are secured, or removing the exceptions from the code.

## Manual Review

After manually going through the code that has been developed so far, I see multiple security issues that will lead to exploits. Hackers will be able to attack the vulnerable spots and gain sensitive information from the company and its users.

The first thing that I noticed is that not all of the methods have assigned getter and setter functions. These are important to protect the data when creating classes. The function of the get method is to return the value that was set or updated by the set method. This is quite an important feature because we can check if the data is valid before setting a property of a class. Another security issue is the properties of the variables. I noticed that most of the variables are set to private, which helps with not leaking the variable information, but one variable in particular caught my eye as it was not set to private. It is a variable within the customer class called *account\_balance*. This is a hazard, as attackers can get hold of the customer's account balance and use this information in their favor. Also, I do not see any security feature around the *showInfo()* function that displays the user account information. This is another risk to security, as it will get exploited to get the information from the customer by an unwanted user. Another issue I noticed was that when starting the Maven dependency check, its version was outdated. The version of Artemis Financial was 5.3.0, compared to the newest available version, 8.1.2. This would also be a critical issue, as we would not be able to catch some of the vulnerabilities as they were not discovered in the previous versions. As one of the Artemis Financial requirements indicates in 1. Interpreting Client Needs, they want to use the most current and effective software security.

## Static Testing

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Vulnerable Dependency | Vulnerability Code | Description | Recommended Solutions | How was this Vulnerability Identified |
| 1 | bcprov-jdk15on-1.46.jar | [**CVE-2016-1000338**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2016-1000338)  [**CVE-2016-1000342**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2016-1000342)  [**CVE-2016-1000343**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2016-1000343)  [**CVE-2016-1000344**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2016-1000344)  [**CVE-2016-1000352**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2016-1000352)  [**CVE-2016-1000341**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2016-1000341)  [**CVE-2016-1000345**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2016-1000345)  [**CVE-2017-13098**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2017-13098)  [**CVE-2020-15522**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-15522)  [**CVE-2016-1000339**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2016-1000339)  **CVE-2020-26939**  **CVE-2015-6644**  [**CVE-2015-7940**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2015-7940)  [**CVE-2018-5382**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2018-5382)  [**CVE-2013-1624**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2013-1624)  [**CVE-2016-1000346**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2016-1000346) | This 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. | Upgrade to the newest version of the Bouncy Castle Cryptography as well as JDK18 as the newest version is only supported on Java Development Kit 18. | Regular Security and Vulnerability checks. |
| 2 | hibernate-validator-6.0.18.Final.jar | [**CVE-2020-10693**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-10693) | Hibernate's Bean Validation (JSR-380) reference implementation. | Attackers can bypass Input validation. It is recommended to upgrade to the newest version. | Regular Security and Vulnerability checks. |
| 3 | jackson-databind-2.10.2.jar | [**CVE-2020-25649**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-25649)  [**CVE-2020-36518**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-36518)  [**CVE-2022-42003**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-42003)  [**CVE-2022-42004**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-42004) | General data-binding functionality for Jackson: works on core streaming API | Data Integrity, Denial of Service, or Deserialization of Untrusted Data. It is recommended to upgrade to the newest version. | Regular Security and Vulnerability checks. |
| 4 | log4j-api-2.12.1.jar | [**CVE-2020-9488**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-9488) | The Apache Log4j API | Threat of SMTPS connection to be intercepted by a man-in-the-middle attack. Recommended to upgrade. | Regular Security and Vulnerability checks. |
| 5 | logback-core-1.2.3.jar | [**CVE-2021-42550**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-42550) | logback-core module | Attacker can execute arbitrary code. Recommended to upgrade. | Regular Security and Vulnerability checks. |
| 6 | snakeyaml-1.25.jar | **CVE-2022-1471**  [**CVE-2017-18640**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2017-18640)  [**CVE-2022-25857**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-25857)  [**CVE-2022-38749**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-38749)  [**CVE-2022-38751**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-38751)  [**CVE-2022-38752**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-38752)  [**CVE-2022-41854**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-41854)  [**CVE-2022-38750**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-38750) | YAML 1.1 parser and emitter for Java | Deserializing yaml content leads to remote code execution and Denial of Service. SafeConsturctor and upgrade are recommended. | Regular Security and Vulnerability checks. |
| 7 | spring-boot-2.2.4.RELEASE.jar | [**CVE-2022-27772**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-27772) | Spring Boot | Vulnerable to temporary directory hijacking. Upgrade to the newest version. | Regular Security and Vulnerability checks. |
| 8 | spring-boot-starter-web-2.2.4.RELEASE.jar | [**CVE-2022-27772**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-27772) | Starter for building web, including RESTful, applications using Spring MVC. Uses Tomcat as the default embedded container | Vulnerable to temporary directory hijacking. Upgrade to the newest version. | Regular Security and Vulnerability checks. |
| 9 | spring-core-5.2.3.RELEASE.jar | [**CVE-2022-22965**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-22965)  [**CVE-2021-22118**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-22118)  [**CVE-2020-5421**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-5421)  [**CVE-2022-22950**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-22950)  [**CVE-2022-22971**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-22971)  [**CVE-2022-22968**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-22968)  [**CVE-2022-22970**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-22970)  [**CVE-2021-22060**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-22060)  [**CVE-2021-22096**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-22096) | Spring Core | Vulnerable to remote code execution (RCE) via data binding, privilege escalation, RFD attacks, Denial of Service, and other issues. Recommended to upgrade to the newest version. | CISA Known Exploited Vulnerability!  VMware Spring Framework on 04-04-2022 |
| 10 | spring-web-5.2.3.RELEASE.jar | [**CVE-2016-1000027**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2016-1000027)  [**CVE-2022-22965**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-22965)  [**CVE-2021-22118**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-22118)  [**CVE-2020-5421**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-5421)  [**CVE-2022-22950**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-22950)  [**CVE-2022-22971**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-22971)  [**CVE-2022-22968**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-22968)  [**CVE-2022-22970**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-22970)  [**CVE-2021-22060**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-22060)  [**CVE-2021-22096**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-22096) | Spring Web | Vulnerable to remote code execution (RCE) via data binding, privilege escalation, RFD attacks, Denial of Service, and other issues. Recommended to upgrade to the newest version. | CISA Known Exploited Vulnerability!  VMware Spring Framework on 04-04-2022 |
| 11 | spring-webmvc-5.2.3.RELEASE.jar | [**CVE-2022-22965**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-22965)  [**CVE-2021-22118**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-22118)  [**CVE-2020-5421**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-5421)  [**CVE-2022-22950**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-22950)  [**CVE-2022-22971**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-22971)  [**CVE-2022-22968**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-22968)  [**CVE-2022-22970**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-22970)  [**CVE-2021-22060**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-22060)  [**CVE-2021-22096**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-22096) | Spring Web MVC | Vulnerable to remote code execution (RCE) via data binding, privilege escalation, RFD attacks, Denial of Service, and other issues. Recommended to upgrade to the newest version. | CISA Known Exploited Vulnerability!  VMware Spring Framework on 04-04-2022 |
| 12 | tomcat-embed-core-9.0.30.jar | [**CVE-2020-1938**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-1938)  [**CVE-2020-11996**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-11996)  [**CVE-2020-13934**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-13934)  [**CVE-2020-13935**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-13935)  [**CVE-2020-17527**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-17527)  [**CVE-2021-25122**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-25122)  [**CVE-2021-41079**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-41079)  [**CVE-2022-29885**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-29885)  [**CVE-2022-42252**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-42252)  [**CVE-2020-9484**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-9484)  [**CVE-2021-25329**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-25329)  [**CVE-2021-30640**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-30640)  [**CVE-2022-34305**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-34305)  [**CVE-2021-24122**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-24122)  [**CVE-2021-33037**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-33037)  [**CVE-2019-17569**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2019-17569)  [**CVE-2020-1935**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-1935)  [**CVE-2020-13943**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-13943)  [**CVE-2021-43980**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-43980) | Core Tomcat implementation | Tomcat gives hires trust to AJP connections, Denial of Service, Information exposure, Infinite Looping, HTTP Request Smuggling, Resource exhaustion, escaping of output, Cross-site Scripting, and other issues. Recommended to upgrade to the newest version. | CISA Known Exploited Vulnerability!  Apache Tomcat on 03-03-2022 |
| 13 | tomcat-embed-websocket-9.0.30.jar | [**CVE-2020-1938**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-1938)  [**CVE-2020-8022**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-8022)  [**CVE-2020-11996**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-11996)  [**CVE-2020-13934**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-13934)  [**CVE-2020-13935**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-13935)  [**CVE-2020-17527**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-17527)  [**CVE-2021-25122**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-25122)  [**CVE-2021-41079**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-41079)  [**CVE-2022-29885**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-29885)  [**CVE-2022-42252**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-42252)  [**CVE-2020-9484**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-9484)  [**CVE-2021-25329**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-25329)  [**CVE-2021-30640**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-30640)  [**CVE-2022-34305**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-34305)  [**CVE-2021-24122**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-24122)  [**CVE-2021-33037**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-33037)  [**CVE-2019-17569**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2019-17569)  [**CVE-2020-1935**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-1935)  [**CVE-2020-13943**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-13943)  [**CVE-2021-43980**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-43980) | Core Tomcat implementation | Tomcat gives hires trust to AJP connections, Incorrect Default Permissions, Denial of Service, Information exposure, Infinite Looping, HTTP Request Smuggling, Resource exhaustion, escaping of output, Cross-site Scripting, and other issues. Recommended to upgrade to the newest version. | CISA Known Exploited Vulnerability!  Apache Tomcat on 03-03-2022 |

## Mitigation Plan

The Manual Review and Static Testing were done and from my excessive security checks, there is a lot of space for improvement. First, I will recommend changes to the code (see the [3. Manual Review](#_Manual_Review)). Variables should be set to private, adding get/ set functions, incorporate security around the *showInfo()* method, update the version of Maven Dependency-check. Second, there are the dependencies that been flagged as security and vulnerability issues (see the [4. Static Testing](#_Static_Testing)). The ones that were marked as critical should be mitigated as soon as possible. These are spring-core-5.2.3.RELEASE.jar, spring-web-5.2.3.RELEASE.jar, spring-webmvc-5.2.3.RELEASE.jar, tomcat-embed-core-9.0.30.jar, tomcat-embed-websocket-9.0.30.jar (for more information see the stage 4. Static Testing, from number 9 to number 13 and click on the vulnerability codes). All of the dependencies are outdated and with exploited vulnerabilities from the past. Updating to the latest versions is required and highly recommended. Also updating to a newer version of JDK is recommended as the new dependencies run in the newest JDK version.

## Action List

* Upgrade Java JDK to the newest version
* Update all dependencies to the newest version
* Change the Maven Dependency-check to run in the newest version
* Set variables to private
* Add get/set functions
* Develop security features for functions and input/ output
* Develop error handling functions
* Create test cases

## References

*North, S. (n.d.). Project One requirements and rubric. Retrieved March 19, 2023, from https://learn.snhu.edu/d2l/le/content/1271170/viewContent/23152571/View*

*Search CVE list*. CVE. (n.d.). Retrieved March 11, 2023, from https://cve.mitre.org/cve/search\_cve\_list.html