****

# 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** | **7/12/2023** | **Alex Roberts** |  |

## 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

Alex Roberts

## Interpreting Client Needs

At GlobalRain we understand that secure communications are vital to your company and your web application. As you are consulting customers with their individualized financial plans you are dealing with important and personal information. Working with savings, retirement, investments, and insurance comes with plenty of security risks as it is. There is a lot of information flowing through these transactions and we need to do everything we can to make sure that nobody outside the circle of that conversation gets access to that information. GlobalRain is an international company so there is a good chance that Artemis Financial is also dealing with international transactions and if that is the case then there are many vulnerabilities and security issues with that alone. There are hackers all over the world looking for doorways into programs to get information or easy money. There could be potential government restrictions about secure communications to consider especially since we are dealing with trade secrets. We would have to ensure that there is no exposure to any of this information. With hackers always on the prowl we need to be vigilant in our security. There is always potential risks when dealing with personal information. If this information is exposed, leaked, or dug up it could be detrimental to everybody involved. Most importantly it could cause havoc or even ruin the customers life if someone were to get access to their bank accounts or steal their identity. Just as well, it could be detrimental to your company in several ways. They could also get access to your information, you could lose a customer, or the customer could pursue a lawsuit against you or us. No matter how we look at it we have to do everything we can to ensure that this does not happen and everybody is safe in the situation. If we are to use open-source libraries, and in most cases we will, we have to ensure that we are using secure sources. As we are a company that deals with this specific work we are knowledgeable about this and will make sure that we are using secure sources and monitor the code that we use. Over time, web applications evolve and become more advanced so we need to make sure that we r keeping up with the most current security processes and also the most current hacker strategies. A lot of research and updating the code we write will be vital to this. This is very important because the more the web evolves the more ways hackers have to get in but also the more ways we have to secure our applications. It’s a never ending battle but we must persevere.

## Areas of Security

These are the areas of security that I believe we need to focus on.

Input validation – RESTful accepts user input so this input from the user needs to be secure. The input should be validated every time a user is on the web application to ensure that the person using the account is the correct person and not someone trying to hack in and steal information.

API’s - The web application is using a RESTful API that is communicating between the server and the user. We need to make sure that this line of communication is secure and information is not being leaked or exposed and getting into the wrong hands.

Code error – We need to ensure that we are handling code errors properly and taking care of them. It’s important to be vigilant in this as errors in the code not only could make the application crash but also leave open vulnerabilities for malicious people to access.

Code Quality – I feel that this is important to any program. It is vital to make sure that you are using proper code techniques along with clean and precise code. This ensures that if there is a problem we can find and fix it easier and leaves less room for errors.

## Manual Review

Maven current version is set to 4.0 but the newest version of maven is actually 3.9.3.

The spring framework is also out of date. The version is set to 2.2.4 release but the current version is 6.0.2.

There is a request parameter with a value of “business names” in the CRUD controller.

The version for JSON is not implemented so it is not clear what to expect from this dependency.

## Static Testing

This is what my dependency check output.

hibernate-validator-6.0.18.Final.jar (pkg:maven/org.hibernate.validator/hibernate-validator@6.0.18.Final, cpe:2.3:a:redhat:hibernate\_validator:6.0.18:\*:\*:\*:\*:\*:\*:\*) : CVE-2020-10693

jackson-databind-2.10.2.jar (pkg:maven/com.fasterxml.jackson.core/jackson-databind@2.10.2, cpe:2.3:a:fasterxml:jackson-databind:2.10.2:\*:\*:\*:\*:\*:\*:\*, cpe:2.3:a:fasterxml:jackson-modules-java8:2.10.2:\*:\*:\*:\*:\*:\*:\*) : CVE-2020-25649, CVE-2020-36518, CVE-2021-46877, CVE-2022-42003, CVE-2022-42004, CVE-2023-35116

log4j-api-2.12.1.jar (pkg:maven/org.apache.logging.log4j/log4j-api@2.12.1, cpe:2.3:a:apache:log4j:2.12.1:\*:\*:\*:\*:\*:\*:\*) : CVE-2020-9488

logback-core-1.2.3.jar (pkg:maven/ch.qos.logback/logback-core@1.2.3, cpe:2.3:a:qos:logback:1.2.3:\*:\*:\*:\*:\*:\*:\*) : CVE-2021-42550

mongo-java-driver-2.4.jar (pkg:maven/org.mongodb/mongo-java-driver@2.4, cpe:2.3:a:mongodb:java\_driver:2.4:\*:\*:\*:\*:\*:\*:\*) : CVE-2021-20328

snakeyaml-1.25.jar (pkg:maven/org.yaml/snakeyaml@1.25, cpe:2.3:a:snakeyaml\_project:snakeyaml:1.25:\*:\*:\*:\*:\*:\*:\*) : CVE-2022-1471, CVE-2017-18640, CVE-2022-25857, CVE-2022-38749, CVE-2022-38751, CVE-2022-38752, CVE-2022-41854, CVE-2022-38750

spring-boot-2.2.4.RELEASE.jar (pkg:maven/org.springframework.boot/spring-boot@2.2.4.RELEASE, cpe:2.3:a:vmware:spring\_boot:2.2.4:release:\*:\*:\*:\*:\*:\*) : CVE-2022-27772, CVE-2023-20883

spring-boot-starter-web-2.2.4.RELEASE.jar (pkg:maven/org.springframework.boot/spring-boot-starter-web@2.2.4.RELEASE, cpe:2.3:a:vmware:spring\_boot:2.2.4:release:\*:\*:\*:\*:\*:\*, cpe:2.3:a:web\_project:web:2.2.4:release:\*:\*:\*:\*:\*:\*) : CVE-2022-27772, CVE-2023-20883

spring-core-5.2.3.RELEASE.jar (pkg:maven/org.springframework/spring-core@5.2.3.RELEASE, cpe:2.3:a:pivotal\_software:spring\_framework:5.2.3:release:\*:\*:\*:\*:\*:\*, cpe:2.3:a:springsource:spring\_framework:5.2.3:release:\*:\*:\*:\*:\*:\*, cpe:2.3:a:vmware:spring\_framework:5.2.3:release:\*:\*:\*:\*:\*:\*) : 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

spring-web-5.2.3.RELEASE.jar (pkg:maven/org.springframework/spring-web@5.2.3.RELEASE, cpe:2.3:a:pivotal\_software:spring\_framework:5.2.3:release:\*:\*:\*:\*:\*:\*, cpe:2.3:a:springsource:spring\_framework:5.2.3:release:\*:\*:\*:\*:\*:\*, cpe:2.3:a:vmware:spring\_framework:5.2.3:release:\*:\*:\*:\*:\*:\*, cpe:2.3:a:web\_project:web:5.2.3:release:\*:\*:\*:\*:\*:\*) : 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

spring-webmvc-5.2.3.RELEASE.jar (pkg:maven/org.springframework/spring-webmvc@5.2.3.RELEASE, cpe:2.3:a:pivotal\_software:spring\_framework:5.2.3:release:\*:\*:\*:\*:\*:\*, cpe:2.3:a:springsource:spring\_framework:5.2.3:release:\*:\*:\*:\*:\*:\*, cpe:2.3:a:vmware:spring\_framework:5.2.3:release:\*:\*:\*:\*:\*:\*, cpe:2.3:a:web\_project:web:5.2.3:release:\*:\*:\*:\*:\*:\*) : 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

tomcat-embed-core-9.0.30.jar (pkg:maven/org.apache.tomcat.embed/tomcat-embed-core@9.0.30, cpe:2.3:a:apache:tomcat:9.0.30:\*:\*:\*:\*:\*:\*:\*, cpe:2.3:a:apache\_tomcat:apache\_tomcat:9.0.30:\*:\*:\*:\*:\*:\*:\*) : CVE-2020-1938, CVE-2020-11996, CVE-2020-13934, CVE-2020-13935, CVE-2020-17527, CVE-2021-25122, CVE-2021-41079, CVE-2022-29885, CVE-2022-42252, CVE-2020-9484, CVE-2021-25329, CVE-2021-30640, CVE-2022-34305, CVE-2021-24122, CVE-2021-33037, CVE-2019-17569, CVE-2020-1935, CVE-2020-13943, CVE-2023-28708, CVE-2021-43980

tomcat-embed-websocket-9.0.30.jar (pkg:maven/org.apache.tomcat.embed/tomcat-embed-websocket@9.0.30, cpe:2.3:a:apache:tomcat:9.0.30:\*:\*:\*:\*:\*:\*:\*, cpe:2.3:a:apache\_tomcat:apache\_tomcat:9.0.30:\*:\*:\*:\*:\*:\*:\*) : CVE-2020-1938, CVE-2020-8022, CVE-2020-11996, CVE-2020-13934, CVE-2020-13935, CVE-2020-17527, CVE-2021-25122, CVE-2021-41079, CVE-2022-29885, CVE-2022-42252, CVE-2020-9484, CVE-2021-25329, CVE-2021-30640, CVE-2022-34305, CVE-2021-24122, CVE-2021-33037, CVE-2019-17569, CVE-2020-1935, CVE-2020-13943, CVE-2023-28708, CVE-2021-43980

## Mitigation Plan

Update maven and spring network as well as anything else inside of the program to ensure that we are using the most current versions.

Remove the request for business names in CRUD as this could potentially cause leaks in the information.

Implement the most current version of JSON to make sure that we know what we are dealing with.

Update the current list of dependencies to make sure we have eyes on all potential vulnerablities.

Go through all of the code to make sure that everything is up to date with the standards of quality code writing.