

# 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** | **03/21/23** | **Isaac Medina** |  |

## 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

B. Isaac Medina

## Interpreting Client Needs

The client is seeking to modernize their practices specifically in the area of security. As with any company, communications need to be secure. Because this is a financial institution there are government specifications on security as well. A couple examples are the Sarbanes-Oxley (SOX) act of 2002, and the Gramm–Leach–Bliley Act (GLBA). These requirements cover things like communication security.

## Areas of Security

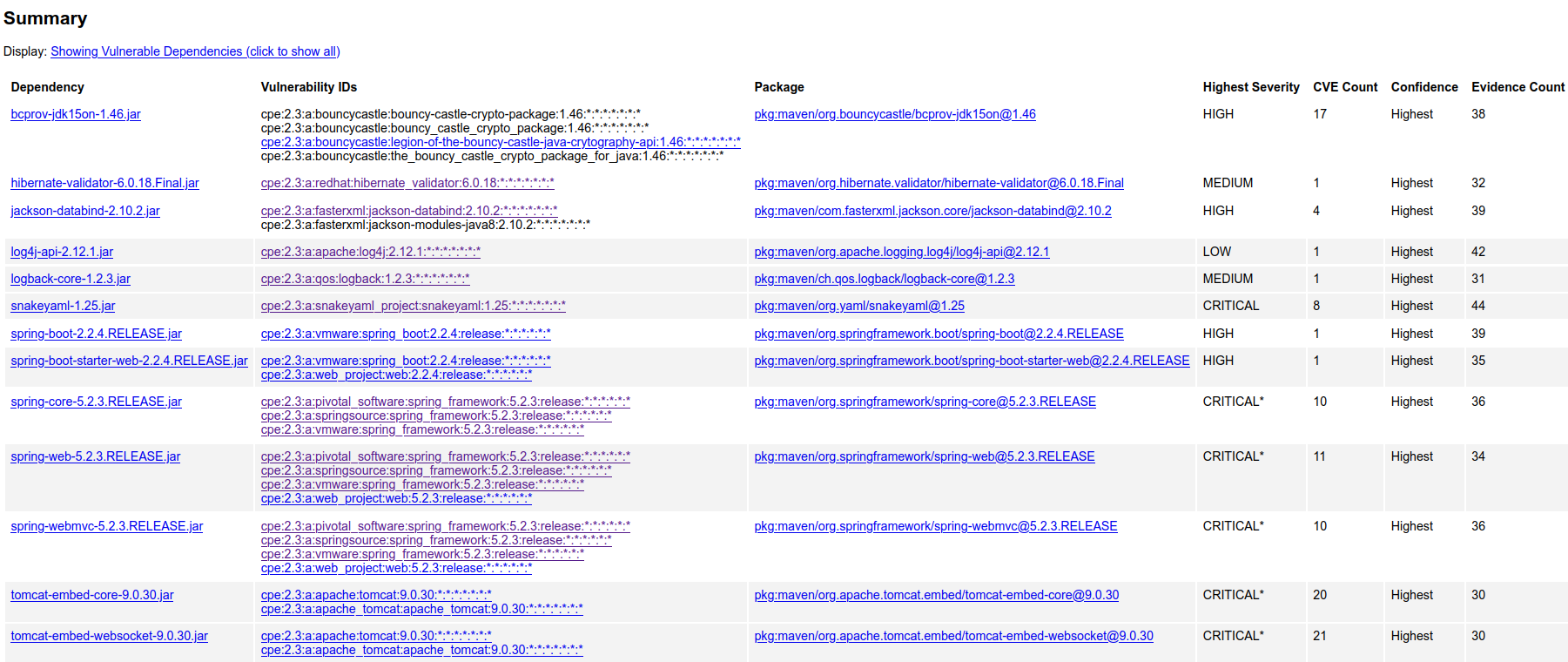
Because we are dealing with a web API there are a few specific focal points we should cover in our assessment. These may include input validation testing, API testing for secure interactions, testing for secure client/server interactions, and of course there should be testing for good coding practices. These testing points will ensure secure communications which are important when transferring private data.

## Manual Review

If I’m correctly interpreting what’s happening, DocData.java is a serious problem. The file has the database name, or at least the port that is used to connect to it, as well as parameters for a username and password, hardcoded into the public ‘read\_document’ method. If the method fails, these fields appear to be a part of the auto generated stack trace.

There are several instances such as in the MavenWrapperDownloader.java file, where there are fields that seem to contain passwords and usernames. These are potentially hazardous but are probably fine as they appear to be used for authentication.

## Static Testing

Attached above is a picture of all the dependencies that have proven vulnerabilities. A summary of the vulnerabilities was a part of the console output at runtime, and it’s contents are as follows.

bcprov-jdk15on-1.46.jar (pkg:maven/org.bouncycastle/bcprov-jdk15on@1.46, cpe:2.3:a:bouncycastle:bouncy-castle-crypto-package:1.46:\*:\*:\*:\*:\*:\*:\*, cpe:2.3:a:bouncycastle:bouncy\_castle\_crypto\_package:1.46:\*:\*:\*:\*:\*:\*:\*, cpe:2.3:a:bouncycastle:legion-of-the-bouncy-castle-java-crytography-api:1.46:\*:\*:\*:\*:\*:\*:\*, cpe:2.3:a:bouncycastle:the\_bouncy\_castle\_crypto\_package\_for\_java:1.46:\*:\*:\*:\*:\*:\*:\*) : CVE-2016-1000338, CVE-2016-1000342, CVE-2016-1000343, CVE-2016-1000344, CVE-2016-1000352, CVE-2016-1000341, CVE-2016-1000345, CVE-2017-13098, CVE-2020-15522, CVE-2020-0187, CVE-2016-1000339, CVE-2020-26939, CVE-2015-7940, CVE-2018-5382, CVE-2013-1624, CVE-2016-1000346, CVE-2015-6644

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-2022-42003, CVE-2022-42004

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

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

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

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-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-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-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

Based off of my examination of all of the vulnerabilities, I can thoroughly say that in all cases it is best to use the most up to date systems. These vulnerabilities all exist strictly due to the fact that the dependencies are out of date. Per the advice from the last vulnerability test, I ran the dependency-updates-report command and was given a monstrous list of all the updates that are available for all the dependencies. I will be submitting that in addition to this report to support my mitigation plan. Figuring out how to get this one command to run took me an eternity but I feel it was well worth it.