****

# CS 305 Project One

**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** | **01/22/2021** | **Hunter Ewin** |  |

## Client



## Instructions

Deliver this completed vulnerability assessment report, identifying your findings of security vulnerabilities and articulating recommendations for next steps to remedy the issues you have found.

Respond to the five steps outlined below and include your findings. Replace the bracketed text on all pages with your own words. If you choose to include images or supporting materials, be sure to insert them throughout.

## Developer

Hunter Ewin

## 1. Interpreting Client Needs

Determine your client’s needs and potential threats and attacks associated with their application and software security requirements. Consider the following regarding how companies protect against external threats based on the scenario information:

Artemis Financial values a very secure environment for their clients. If this environment was breached it would jeopardize its operations. This application is for conducting financial transactions along with financial transfers. This information is very sensitive and confidential. A breach of this information would result in loss of business along with hurting the reputation of this application. Artemis Financial could also face certain punishments or penalties for failing to keep this information secure.

Artemis Financial produces many international transactions. Such transactions may involve internetworking of customers through international ways. With that being said, there is also a possibility of these customers traveling to different places around the world while still needing access to their information.

There are multiple government restrictions when it comes to making communication secure. When dealing with certain transactions with European countries, there is a policy called General Data Protection Regulation. This policy regulates communications by requiring that the owners and service providers ensure their information is protected or secured. Which is why Artemis Financial should ensure there are zero loopholes in security.

Some external threats would include threats such as authentication, DOS (denial of service), API, and error handling. These threats could potentially hand over its users information.

Incremental modernization would be one of the requirements in that it should be done in an incremental way in order to reduce the operation risks of legacy and open source libraries. Modernization should be done by making these systems accessible. This doing so will allow upgrades or any required changes to be done in a timely manner. It will also assist by helping the improvement of the security functions.

## 2. Areas of Security

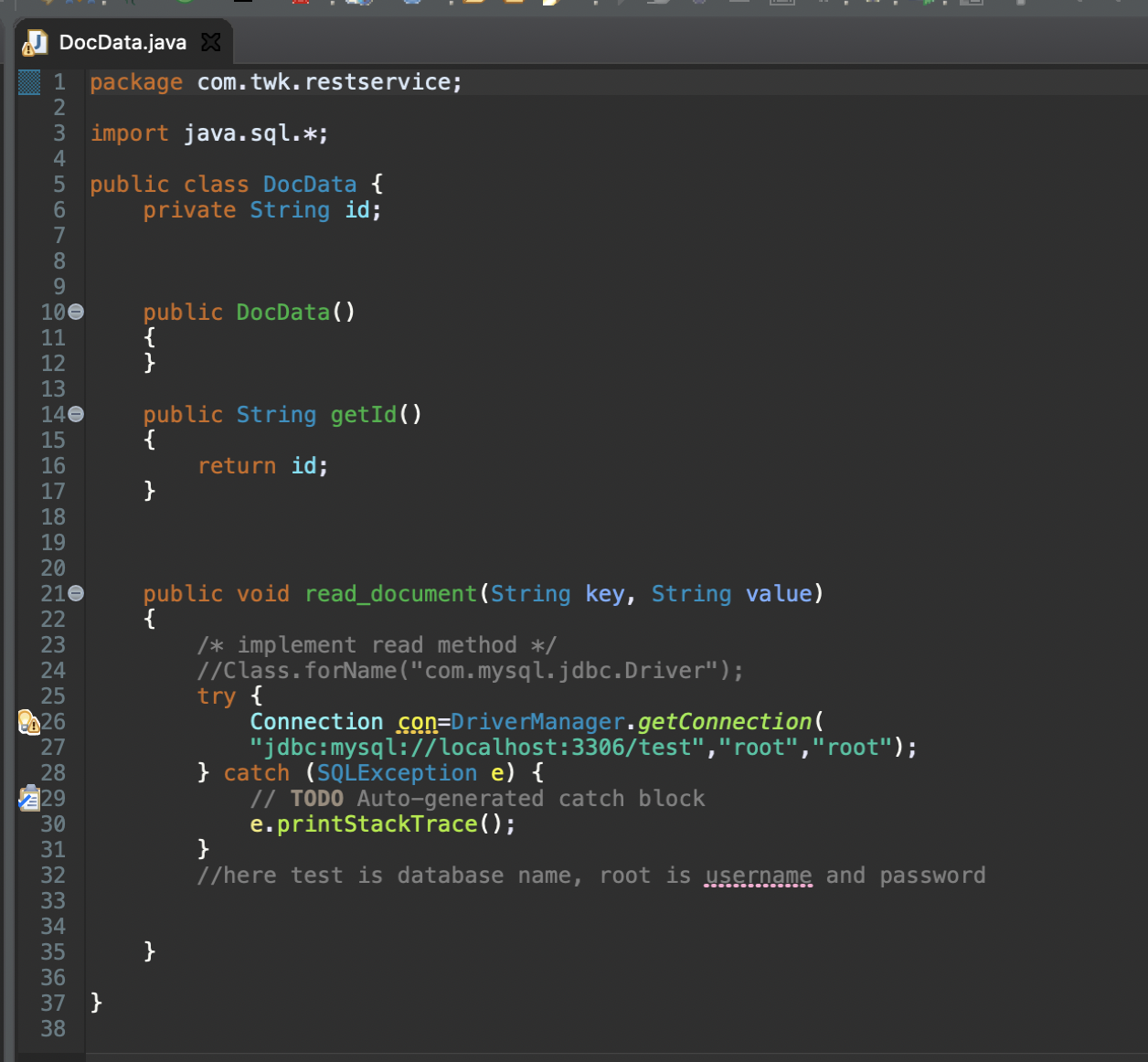
API – There needs to be a secure configuration for the APIs in order to protect the security between the system and the API connection.

Cryptography – This would be the access of the information the client receives over the web. If this is not properly encrypted, it could result in loss of integrity. The company should use proper certificate validation.

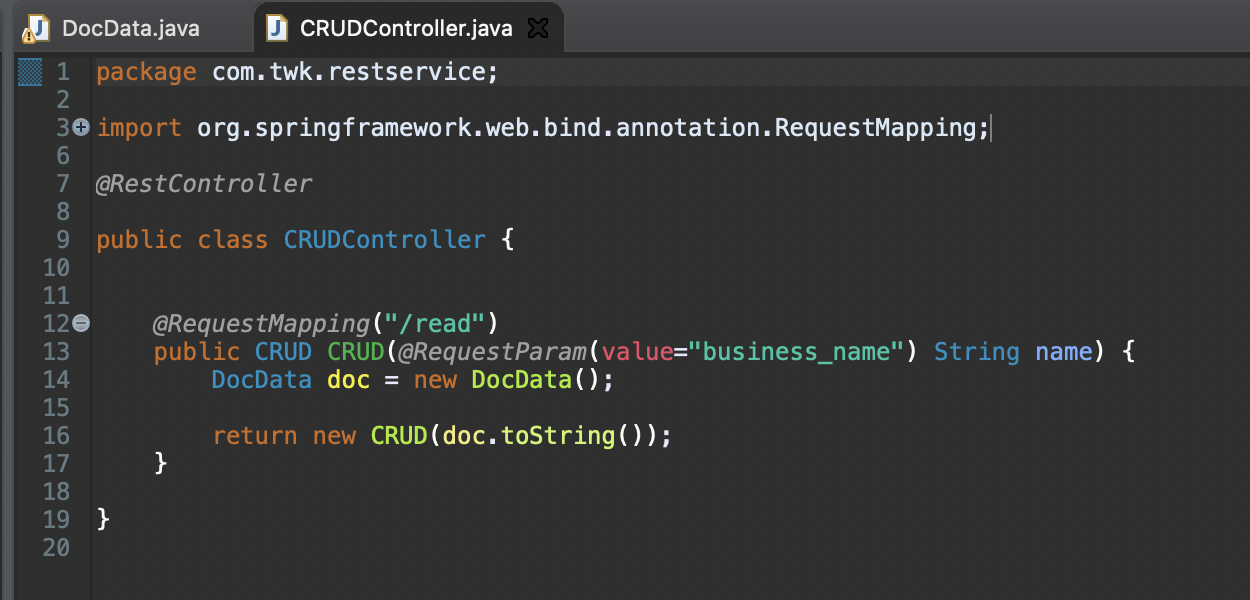
Client/server – This would apply due to the customer wanting to receive information from the servers that hold their information to provide transactions. This is very important in the fact that the end users information will be traveling over the web.

## 3. Manual Review

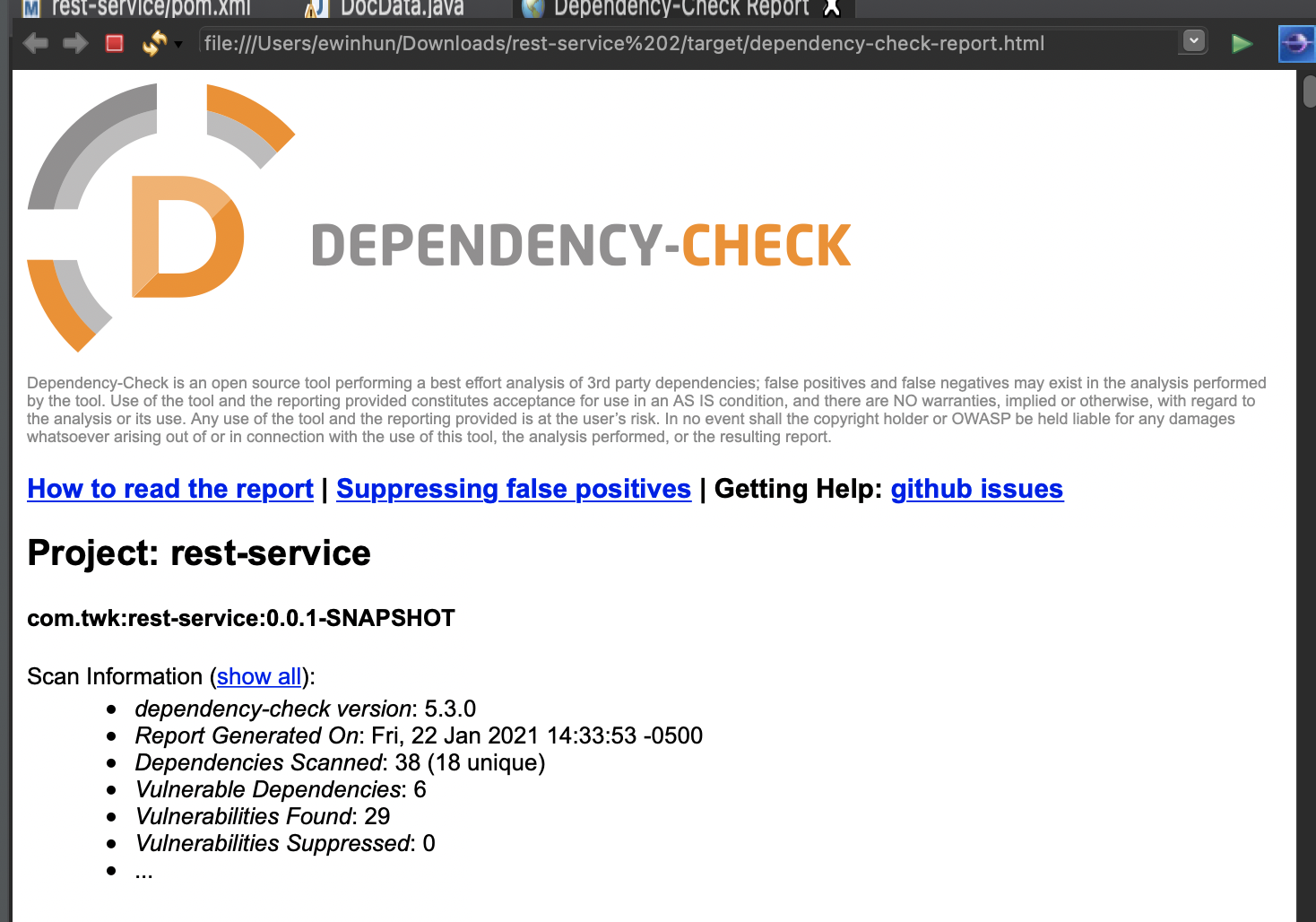
The DocData.java has a vulnerability concerning data access. This method involves the location of the database and the username and password. There is a root user and password, which are not recommended to use, but it can be easily hacked. This would make it easy for a hacker to access the system.

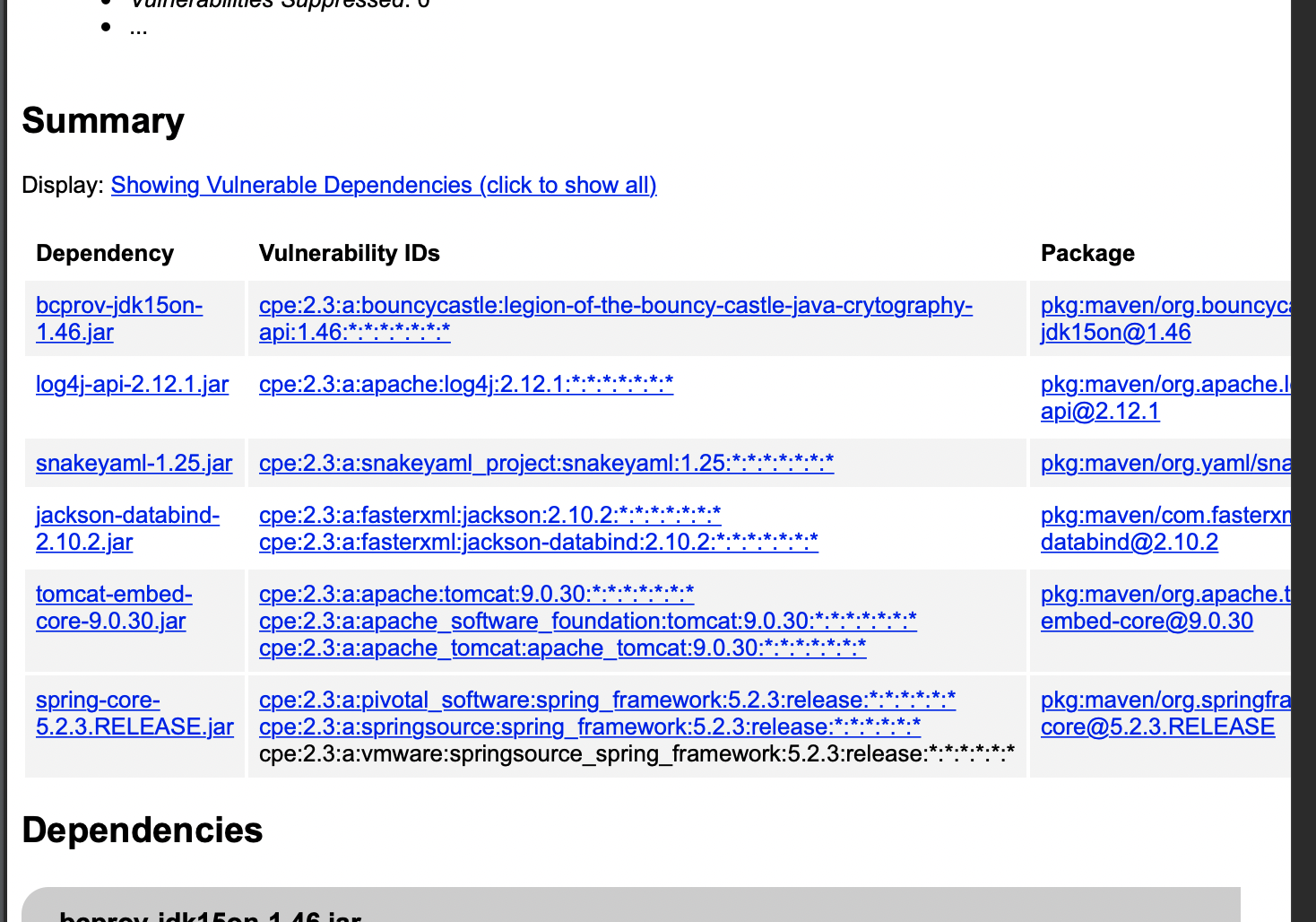


CRUDController.java shows a vulnerability where the application could be exposing internal objects. This could expose the docdata object in the database acess.



## 4. Static Testing





## 5. Mitigation Plan

Access to username and password must be properly created. The password should have certain limitations that have to be met in order to be secure. Characters, numbers, and letters all much be included as well as a certain length.

Code quality should be able to be modified in order for proper authentication and catching certain errors. This will help in preventing attacks or giving away user information.

It is very important to apply updates to the Apache server to the most recent version. This will prevent the CVE-2020-9488 vulnerability. Updates from the previous versions should fix this.

The different cryptographic certificates should be validated. This will prevent the CVE-2020-9488, and the CVE-2020-13935 vulnerabilities.

There should also be an implementation of a check-in for both the client and server sides certificates. This will prevent the associated client API requests and authentication procedures.