****

# 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** | **03/28/2021** | **Eusra Shams** |  |

## 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

Eusra Shams

## 1. Interpreting Client Needs

Artemis Financial provides financial planning for clients, which includes conducting financial transactions and transfers, as well as holding multiple types of financial accounts (savings, retirement, investment, etc.) for clients. It is imperative that the communication for each of these accounts and transactions remains secure and private between each financial advisor and their individual client.

The company has the potential to conduct a multitude of international transactions, as clients or financial advisors may choose to invest in international markets. Additionally, clients may be in various locations around the world or conduct transactions while travelling internationally.

Governmental restrictions vary greatly, depending on the region of the world. While the US does not have regulations that affect the whole country, California has the CCPA1 of 2018. Countries in the European Union are also under the GDPR2 policy that regulates communications. It is necessary to adhere to the strictest guidelines available to ensure all communications are secured to the standards necessary to conduct international transactions.

API security risks are present that may include:3

* Authentication of user or function – proper identification of user or function
* Not handling authorization separate from authentication – checking whether user or function can access given resource
* Data exposure – excess exposure to unauthorized users
* Insufficient logging and monitoring – same vulnerabilities attacked multiple times
* Security misconfigurations
* DDoS attacks – excess traffic to make application crash
* Insufficient server security – unsecured web-based or app based access or requests are accepted

## 2. Areas of Security

* API – the web-based application will require secured configuration to prevent any compromise of security between the system and API connection.
* Cryptography – data as transactions or account information will be transmitted. It is necessary to ensure proper encryption to prevent exposure to unauthorized users or attackers.
* Client/Server – the clients here are the users Artemis financial will be servicing, who will be interacting with the server to conduct financial transactions, and retrieve account information. It will be necessary to ensure connections are secure and private.
* Secure coding – it is necessary to ensure the code is valid and up to the security standards as required by multiple governing bodies, as well as provide testing against multiple types of attacks or errors.
* Encapsulation – account information and user data are stored as databases, and will need to be protected from unauthorized access or modifications.

## 3. Manual Review

* CRUDController.java has the CRUD method that is accessed by passing “business\_name” to return the DocData object. Here, code can be injected to gain access to the database.

**public** CRUD CRUD(@RequestParam(value="business\_name") String name) {

DocData doc = **new** DocData();

**return** **new** CRUD(doc.toString());

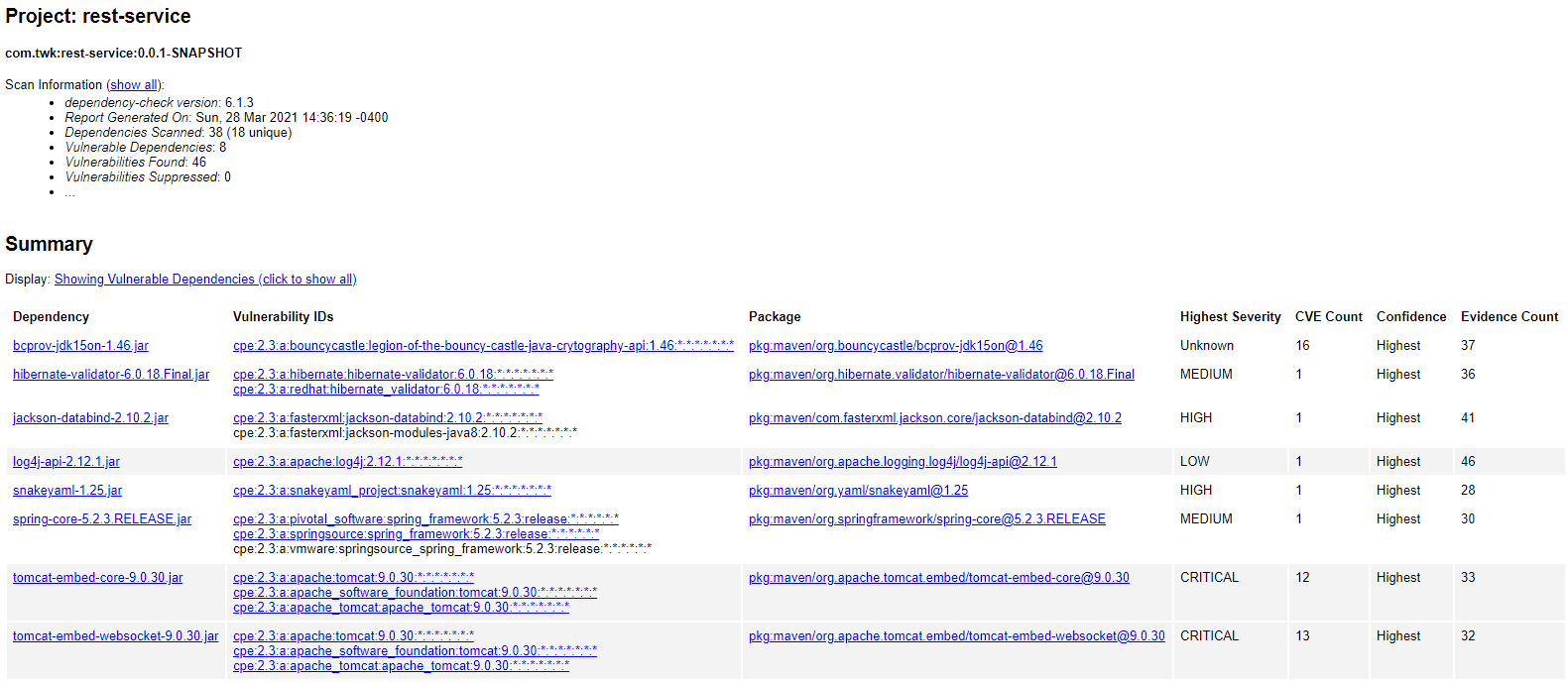
}

* DataDoc.java shows vulnerability to data access as these are easy to find by unauthorized users through use of brute force attacks.

## 4. Static Testing

Run a dependency check on Artemis Financial’s software application to identify all security vulnerabilities in the code. Record the output from dependency check report. Include the following:

1. The names or vulnerability codes of the known vulnerabilities
2. A brief description and recommended solutions provided by the dependency check report
3. Attribution (if any) that documents how this vulnerability has been identified or documented previously



|  |  |  |
| --- | --- | --- |
| **Vulnerability Code (CVE)** | **Descriptor** | **Recommended solutions** |
| 2013-1624 | Implemented TLS has no proper consideration on timing side channel attacks. | Mitigation strategies are required for proper timing checks, and to prevent timing data attacks. |
| 2015-6644 | Inappropriate information disclosure may occur by accessing private data stored in database. | The application will need to be updated regularly. |
| 2017-13098 | Provides a weak Bleichenbacher oracle when any TLS cipher suite using RSA key exchange is negotiated. An attacker can recover the private key from a vulnerable application. | Listed mitigation strategy is to disable TLS RSA cyphers if possible for affected users and administrators, and apply updates |
| 2020-9488 | There are associated certificate validations in Apache SMTP appender. This may allow for man-in-the-middle attacks that cause intercepted connections. | Upgrades to SmtpAppender 2.13.2 will all host name to match with the SSL/TLS certificate of SMTPS connection. |
| 2020-13934 | OutOfMemoryException could occur leading to a denial of service. | Upgrades to Apache Tomcat 10.0.0-M7 or later is the related mitigation strategy. |
| 2020-13935 | Invalid payload lengths could trigger an infinite loop. Multiple requests with invalid payload lengths could lead to a denial of service. | Validation measures will need to be put into place to prevent errors. |
| 2020-1938 | When using the Apache JServ Protocol (AJP), care must be taken when trusting incoming connections to Apache Tomcat. Tomcat treats AJP connections as having higher trust than, for example, a similar HTTP connection. If such connections are available to an attacker, they can be exploited | Update to Tomcat 9.0.31 will resolve the issues. |
| 2019-17569 | Invalid Transfer-Encoding headers were incorrectly processed leading to a possibility of HTTP Request Smuggling if Tomcat was located behind a reverse proxy that incorrectly handled the invalid Transfer-Encoding header in a particular manner. | Update to Tomcat 9.0.31 will resolve the issues. |

## 5. Mitigation Plan

The code will need to be modified to require username and password generation that meets multiple security criteria (length, special characters, etc.). The code should also prevent using the same string for username and passwords. The access to the database will require more than injection of “business\_name” to access database.

The Apache server will be updated to the most current version to prevent multiple vulnerabilities. Additionally, the code should be able to handle validation of multiple cryptographic certificates in order to prevent main-in-the-middle attacks.

**References**

1. California Consumer Privacy Act (CCPA). (2021, March 4). State of California - Department of Justice - Office of the Attorney General. <https://oag.ca.gov/privacy/ccpa>
2. General Data Protection Regulation (GDPR) – Official Legal Text. (2019, September 2). General Data Protection Regulation (GDPR). <https://gdpr-info.eu/>
3. Gilling, D. (2020, July 7). Top 10 API Security Threats Every API Team Should Know. Top 10 API Security Threats Every API Team Should Know | Moesif Blog. https://www.moesif.com/blog/technical/api-security/API-Security-Threats-Every-API-Team-Should-Know/