****

# 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** | **3/18/2022** | **Clay Dodson** | **Security Assessment** |

## Client



## Developer

Clay Dodson

## 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:

* What is the value of secure communications to the company? Artemis Financial deals with highly sensitive information as a part of their daily business. Communications sent between the company and clients will often contain this sensitive information. A breach of this data could result in loss of income/livelihood for clients which could mean a lawsuit for the company. Therefore, secure communications are of the highest priority for a financial institution. These communications must have encryption, secure transfer, and user authentication to name a few security protocols.
* Are there any international transactions that the company produces? While not explicitly mentioned in the scenario, it can be assumed that Artemis Financial will deal with international transactions as clients may wish to transfer funds to an international account. The company may also have clients in other countries.
* Are there governmental restrictions about secure communications to consider? The Office of the Comptroller of the Currency (OCC), a branch of the U.S. Department of the Treasury regulates and oversees banks and their operations. The OCC’s mission is “to ensure that national banks and federal savings associations operate in a safe and sound manner, provide fair access to financial services, treat customers fairly, and comply with applicable laws and regulations.” With this government oversight, there are laws and regulations about secure communications that Artemis Financial will need to follow.
* What external threats might be present now and in the immediate future? Any number of known or unknown vulnerabilities in the software could pose a threat to Artemis Financial as hackers could take advantage of these vulnerabilities. Without proper security, hackers could expose sensitive client data, manipulate account information, install key readers to obtain admin passwords, install viruses, and even reroute a transaction to another account. With mitigation techniques, many of these threats can be reduced and even eliminated.
* What are the “modernization” requirements that must be considered, such as the role of open-source libraries and evolving web application technologies? Artemis Financial plans to modernize its operations by adding layers of security to their RESTful API. This can be done through the use of existing frameworks like Spring Security. Utilizing existing web technologies like HTTPS, open-source algorithms for password hashing, and input validation will further improve the security of the API.

## 2. Areas of Security

Referring to the Vulnerability Assessment Process Flow Diagram, identify which areas of security are applicable to Artemis Financial’s software application. Justify your reasoning for why each area is relevant to the software application.

* Input Validation – User inputs will need to be validated to avoid overflow and other errors which could break the system or expose sensitive data.
* APIs – Artemis Financial’s program runs a representative state transfer (REST) application programming interface. So, any transfers via the API will need to be secure.
* Cryptography – Since the program handles sensitive financial information, then that data will need to be encrypted.
* Client/Server – Artemis Financial’s system will need to interact with other systems and databases and those interactions/transactions will need to be secure and free of vulnerability.
* Encapsulation – Due to the highly sensitive nature of the data being handled within the system, then certain classes, variables, and functions should only be viewable by those with a business need-to-know.

## 3. Manual Review

Continue working through the Vulnerability Assessment Process Flow Diagram. Identify all vulnerabilities in the code base by manually inspecting the code.

* CRUD.java – The CRUD class is a public class with all public functions
* CRUDController.java – The CRUD controller class is a public class and its only function is public
* customer.java – The customer class is a public class, and the show account number function, show account balance functions, and deposit info are all public as well.
* DocData.java – Class and functions are not private. The function to access the SQL data could reveal usernames and passwords.
* Greeting.java – Class and functions are all public potentially revealing variables “id” and “content” which contain sensitive customer data.

## 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 and vulnerability codes of the known vulnerabilities. Only those vulnerabilities with the highest CVE count and severity were included.
2. spring-aop-5.2.3.RELEASE.jar
   * CVE-2020-5421
   * CVE-2021-22060
   * CVE-2021-22096
   * CVE-2021-22118
3. spring-core-5.2.3.RELEASE.jar
   * CVE-2020-5421
   * CVE-2021-22060
   * CVE-2021-22096
   * CVE-2021-22118
4. tomcat-embed-core-9.0.30.jar
   * CVE 2019-17569
   * CVE 2020-11996
   * CVE 2020-13934
   * CVE 2020-13935
   * CVE 2020-13943
   * CVE 2020-17527
   * CVE 2020-1935
   * CVE 2020-1938
   * CVE 2020-9484
   * CVE 2021-24122
   * CVE 2021-25122
   * CVE 2021-25329
   * CVE 2021-30640
   * CVE 2021-33037
   * CVE 2021-41079
   * CVE 2021-42340
5. tomcat-embed-websocket-9.0.30.jar
   * CVE 2019-17569
   * CVE 2020-11996
   * CVE 2020-13934
   * CVE 2020-13935
   * CVE 2020-13943
   * CVE 2020-17527
   * CVE 2020-1935
   * CVE 2020-1938
   * CVE 2020-8022
   * CVE 2020-9484
   * CVE 2021-24122
   * CVE 2021-25122
   * CVE 2021-25329
   * CVE 2021-30640
   * CVE 2021-33037
   * CVE 2021-41079
   * CVE 2021-42340
6. bcprov-jdk15on-1.46.jar
   * CVE 2013-1624
   * CVE 2015-7940
   * CVE 2015-6644
   * CVE 2016-1000338
   * CVE 2016-1000339
   * CVE 2016-1000341
   * CVE 2016-1000342
   * CVE 2016-1000343
   * CVE 2016-1000344
   * CVE 2016-1000345
   * CVE 2016-1000346
   * CVE 2016-1000352
   * CVE 2017-13098
   * CVE 2018-1000613
   * CVE 2018-5382
   * CVE 202-15522
   * CVE 202-26939
7. A brief description and recommended solutions provided by the dependency check report
8. spring-aop-5.2.3.RELEASE.jar and spring-core-5.2.3.RELEASE.jar – These vulnerabilities allow malicious users to read and modify files, provide input, and bypass certain protections. It is recommended to upgrade to the following versions of Spring Framework:1

For 3.2.x upgrade to 3.2.15+.

For 4.0.x and 4.1.x upgrade to 4.1.8+.

For 4.2.x upgrade to 4.2.2+.

1. tomcat-embed-core-9.0.30.jar and tomcat-embed-websocket-9.0.30.jar – These vulnerabilities allow malicious users to utilize HTTP Request Smuggling, push denial of service attacks by triggering infinite loops, leak HTTP data between requests, and gain access to unauthorized privileges. It is recommended to upgrade to version 8.5.54-0+deb9u1 and update all tomcat8 packages.2
2. bcprov-jdk15on-1.46.jar – Cryptographic issue allowing remote users to conduct a series of attacks via timing mechanisms and manipulation of data structures holding sensitive information. Software owner Oracle recommends applying Critical Patch Update fixes as soon as possible to remedy these vulnerabilities.3
3. Attribution (if any) that documents how this vulnerability has been identified or documented previously:
4. CVE-2015-5211 RFD attack in Spring Framework: Security. VMware Tanzu. (n.d.). Retrieved March 21, 2022, from <https://tanzu.vmware.com/security/cve-2015-5211>
5. Debian Security Advisory. Debian. (n.d.). Retrieved March 21, 2022, from <https://www.debian.org/security/2020/dsa-4673>
6. Oracle Critical Patch Update Advisory - April 2018. Oracle Critical Patch Update - April 2018. (n.d.). Retrieved March 21, 2022, from <https://www.oracle.com/security-alerts/cpuapr2018.html>

## 5. Mitigation Plan

After interpreting your results from the manual review and static testing, identify the steps to remedy the identified security vulnerabilities for Artemis Financial’s software application.

1. Apply encapsulation to specific classes, variables, and functions throughout the program. Only public variables and functions should be those containing non-encrypted, non-sensitive data that needs to be utilized by other classes within the program.
2. Apply proper cryptography to all sensitive data.
3. Add functions to validate user input in order to avoid data overloads and protect against known vulnerabilities.
4. Update Spring Framework to newest version
5. Update TomCat to version 8.5.54-0+deb9u1
6. Update all TomCat packages
7. Apply Oracle’s Critical Patch Update for Bouncy Castle
8. Update Bouncy Castle Crypto package