****

# 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** | **7/15/2022** | **Corey Nance** |  |

## 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

Corey Nance

## 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?
* Are there any international transactions that the company produces?
* Are there governmental restrictions about secure communications to consider?
* What external threats might be present now and in the immediate future?
* What are the “modernization” requirements that must be considered, such as the role of open source libraries and evolving web application technologies?

The main type of communication with Artemis Financial would be transferring important financial data back and forth which is gives a great deal of value to secure communications, protecting this type of data is the utmost of importance. Having secure or encrypted communication can help to deter hackers, keep communications confidential and private, and protect against cyber-attacks (Allen, 2022).

In this area, judging by the type of company this is, one would think that at least of the transactions would be international. This area needs to be taken into consideration when thinking of the types of security that is needed.

The main thing the bank needs to do is protect the customers assets and digital banking transactions. The main type of threats a bank may have to face would be unencrypted data, malware, third-party services, spoofing, and phishing (Shaji, 2020). Within these areas, the company has to plan for the threats that are now and the advancements made in these areas in the future

Requirements to be considered would have to with the advancement of applications and their threats where both areas are always evolving. Attackers are more active across all threat types like cloud infrastructures which only state of the art security can protect (Hoffman, 2022). Open-source libraries play a key role because historically, open-source collaboration has helped applications become more secure and scalable (Brooks, 2021).

## 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.

For this program using what has been learned so far, the areas of security that’s applicable to Artmeis Financial’s web application would be Input Validation, APIs, Cryptography, Client/Server, Code Error, and Encapsulation.

Since the application will be taking inputs, those inputs must be checked in validated, one example would be input length restrictions. Checking inputs helps to prevent mistakes, memory leakage, and attacks like injection attacks that can compromise the system (Mezquita, 2021).

APIs are another area mainly because the APIs are usually accessed from outside the application which means that they to have the same risks as inputs. Implementing a RESTful api will help keep the application simple and securing the transport layer, access control by using authentication and rate limiting will help to secure the api (Domoney, n.d). The goal would be to ensure that the api is created so that there is some form of input validation that does not allow unexpected access to the application. That brings up the next area which is Cryptography.

This area is where encryption techniques are used to protect the company and customers data as well as any interactions or sessions. Client/Server along with Encapsulation will help to separate the client side of the application from the server side. This along with encapsulating the data controllers helps to deny any access to important API access functions or data transfers.

Lastly, securely handling any code errors is important with this application so that attackers are not able to gain access by use of the system generated errors. Code not being handled securely can allow for sensitive information to get exposed and an attacker can exploit it (Poston, 2021).

## 3. Manual Review

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

Starting off, the first area would be to update all pom file dependencies to latest versions. After reviewing the main code base, the main area of vulnerability looks to be within the validation areas. With reviewing the CRUD.java file, this looks to be only the server side processes, because of this one would believe or assume that all client-side password encryption is being down as well as user session validation is done.

On the server side, reviewing the CRUDcontroller, a vulnerability that can be seen is that the values coming from the Requestmapping are not checked at all. A validation of business\_name length and not being null would help to add security. With the customer.java file, there looks to be a vulnerability from not verifying the user session before allowing the user to see the balance. There also looks to be another vulnerability in the encapsulation area. The account number is set to private, but the account balance is not. The account balance should also be set to private so that that its encapsulated and access is limited.

In the DocData.java file, one area of vulnerability is that when the ID is retrieved, there are not any forms of validation that the ID is what its suppose to be. Id length checks, char content checks, and not null checks would help to secure any information coming into the class.

With the Greeting.java file, more input validations would help close the vulnerability of when the ID and greeting content is retrieved. It could be hard to verify content other than setting content length limits but the ID is vulnerable to attack if its not checked.

Even though the checks would be with in the greeting class, there is still a vulnerability in the Greeting Controller.java file because there are not any URL validations from the Greeting method before the parameters are used.

The last area of vulnerability would be with the myDateTime.java file where adding checks for date format can help to ensure that the date format is what is expected.

## 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

* **Dependency -** bcprov-jdk15on-1.46.jar
  + - **Vulnerability - CVE-2016-1000352**

The TLS implementation in the Bouncy Castle Java library before 1.48 and C# library before 1.8 does not properly consider timing side-channel attacks on a noncompliant MAC check operation during the processing of malformed CBC padding, which allows remote attackers to conduct distinguishing attacks and plaintext-recovery attacks via statistical analysis of timing data for crafted packets, a related issue to CVE-2013-0169.

* + - **Solution**

Research says that it’s a false positive

source - <https://github.com/jeremylong/DependencyCheck/issues/2501>

* **Dependency -** spring-boot-2.2.4.RELEASE.jar
  + - **Vulnerability - CVE-2022-27772**

The Spring OXM wrapper in Spring Framework before 3.2.4 and 4.0.0.M1, when using the JAXB marshaller, does not disable entity resolution, which allows context-dependent attackers to read arbitrary files, cause a denial of service, and conduct CSRF attacks via an XML external entity declaration in conjunction with an entity reference in a (1) DOMSource, (2) StAXSource, (3) SAXSource, or (4) StreamSource, aka an XML External Entity (XXE) issue.distinguishing attacks and plaintext-recovery attacks via statistical analysis of timing data for crafted packets, a related issue to CVE-2013-0169.

* + - **Solution**

The issue is concerned with XML entity resolution. The idea is to use an XML entity to resolve to an local file on the host system (for instance, /etc/passwd). See https://www.owasp.org/index.php/XML\_External\_Entity\_(XXE)\_Processing for more details. To resolve this issue, simply disable external entity resolution when dealing with XML from external sources. The way to do that depends on the XML API that you use. Update to latest version of spring.

Source - <https://github.com/spring-projects/spring-framework/issues/15432>

* **Dependency -** logback-core-1.2.3.jar
  + - **Vulnerability - CVE-2021-42550**

In logback version 1.2.7 and prior versions, an attacker with the required privileges to edit configurations files could craft a malicious configuration allowing to execute arbitrary code loaded from LDAP servers

* + - **Solution**

In general, the triggering method of the vulnerability described in the following is relatively difficult, unless the following conditions can be met, possible upgrade to version 1.3

Source - <https://security.netapp.com/advisory/ntap-20211229-0001/>

* **Dependency -** log4j-api-2.12.1.jar
  + - **Vulnerability - CVE-2020-9488**

Apache Log4j2 versions 2.0-beta7 through 2.17.0 (excluding security fix releases 2.3.2 and 2.12.4) are vulnerable to a remote code execution (RCE) attack when a configuration uses a JDBC Appender with a JNDI LDAP data source URI when an attacker has control of the target LDAP server. This issue is fixed by limiting JNDI data source names to the java protocol in Log4j2 versions 2.17.1, 2.12.4, and 2.3.2. Published: December 28, 2021; 3:15:08 PM -0500

* + - **Solution**

Upgrade to 2.13.2 which supports this feature. Previous versions can set the system property mail.smtp.ssl.checkserveridentity to true to globally enable hostname verification for SMTPS connections.

Source - https://issues.apache.org/jira/browse/LOG4J2-2819

* **Dependency -** jakarta.annotation-api-1.3.5.jar
  + - **Vulnerability - CVE-2022-31569**

The RipudamanKaushikDal/projects repository through 2022-04-03 on GitHub allows absolute path traversal because the Flask send\_file function is used unsafely. CWE-22 Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')

* + - **Solution –** No results as for solutions found
* **Dependency -** snakeyaml-1.25.jar
  + - **Vulnerability - CVE-2017-18640**

The Alias feature in SnakeYAML 1.18 allows entity expansion during a load operation, a related issue to CVE-2003-1564. CWE-776 Improper Restriction of Recursive Entity References in DTDs ('XML Entity Expansion')

* + - **Solution**

This update can be installed with the "dnf" update program. Use

su -c 'dnf upgrade --advisory FEDORA-2020-23012fafbc' at the command

line. For more information, refer to the dnf documentation available at

source - htt<ps://lists.fedoraproject.org/archives/list/package-ann>ounce@lists.fedoraproject.org/message/PTVJC54XGX26UJVVYCXZ7D25X3R5T2G6/

* **Dependency -** jackson-databind-2.10.2.jar
  + - **Vulnerability - CVE-2020-25649**

A flaw was found in FasterXML Jackson Databind, where it did not have entity expansion secured properly. This flaw allows vulnerability to XML external entity (XXE) attacks. The highest threat from this vulnerability is data integrity.

* + - **Solution**

Numerous software updates to fix this issue

Source - https://security.netapp.com/advisory/ntap-20210108-0007/

* **Dependency -** tomcat-embed-core-9.0.30.jar
  + - **Vulnerability - CVE-2019-17569**

The refactoring present in Apache Tomcat 9.0.28 to 9.0.30, 8.5.48 to 8.5.50 and 7.0.98 to 7.0.99 introduced a regression. The result of the regression was that 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. Such a reverse proxy is considered unlikely.

* + - **Solution**

Upgrade to Apache Tomcat 10.0.0-M6 or later, Upgrade to Apache Tomcat 9.0.36 or later, Upgrade to Apache Tomcat 8.5.56 or later

Source - <https://lists.apache.org/thread/sqyqm0nn7g26bbsd2rm0g4sc3woo32mc>

* **Dependency -** hibernate-validator-6.0.18.Final.jar
  + - **Vulnerability - CVE-2020-10693**

A flaw was found in Hibernate Validator version 6.1.2.Final. A bug in the message interpolation processor enables invalid EL expressions to be evaluated as if they were valid. This flaw allows attackers to bypass input sanitation (escaping, stripping) controls that developers may have put in place when handling user-controlled data in error messages.

* + - **Solution**

This issue has been addressed in the following products: Red Hat JBoss Enterprise Application Platform 7.3 for RHEL 8. Update to new version

Source - <https://bugzilla.redhat.com/show_bug.cgi?id=CVE-2020-10693>

* **Dependency -** spring-core-5.2.3.RELEASE.jar
  + - **Vulnerability - CVE-2016-1000027**

Pivotal Spring Framework through 5.3.16 suffers from a potential remote code execution (RCE) issue if used for Java deserialization of untrusted data. Depending on how the library is implemented within a product, this issue may or not occur, and authentication may be required. NOTE: the vendor's position is that untrusted data is not an intended use case. The product's behavior will not be changed because some users rely on deserialization of trusted data.

* + - **Solution**

Numerous patch updates

Source - https://security.netapp.com/advisory/ntap-20210513-0009/

## 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.

* + - To add input validation throughout the code base to enhance security
    - To update all dependency vulnerabilities to the latest software versions
    - Update pom file dependencies to latest versions.

**References**

Allen, N. (2022, February 14). *7 reasons why your organisation needs secure communications - salt: Secure communications*. Salt. Retrieved July 15, 2022, from https://saltcommunications.com/news/7-reasons-why-your-organisation-needs-secure-communications/

Brooks, C. (2021, January 12). *Open source: A catalyst for Modernization & Innovation*. Servers & Storage. Retrieved July 16, 2022, from https://www.ibm.com/blogs/systems/open-source-a-catalyst-for-modernization-innovation/

Domoney, C. (n.d.). *Rest API security essentials - dzone refcardz*. dzone.com. Retrieved July 16, 2022, from https://dzone.com/refcardz/rest-api-security-1

Hoffman, Y. (2022, March 25). *Application modernization: The security implications of binding old and new together*. Security Boulevard. Retrieved July 16, 2022, from https://securityboulevard.com/2022/03/application-modernization-the-security-implications-of-binding-old-and-new-together/

Mezquita, T. (2021, August 20). *Input validation - Cyberhoot cyber library*. CyberHoot. Retrieved July 16, 2022, from https://cyberhoot.com/cybrary/input-validation/

Poston, H. (2021, August 15). *Improper error handling*. Infosec Resources. Retrieved July 9, 2022, from <https://resources.infosecinstitute.com/topic/improper-error-handling/#:~:text=Handling%20errors%20correctly%20is%20essential,could%20aid%20in%20its%20exploitation>

Reuters, T. (n.d.). *Internet privacy laws Internet privacy laws revealed - how your personal information is protected online*. Internet Privacy Laws Revealed - How Your Personal Information is Protected Online | Thomson Reuters. Retrieved July 16, 2022, from https://legal.thomsonreuters.com/en/insights/articles/how-your-personal-information-is-protected-online

Shaji, A. M. (2020, November 20). *Cybersecurity in digital banking: Threats, challenges and solution*. Enterslice. Retrieved July 16, 2022, from https://enterslice.com/learning/cybersecurity-in-digital-banking-threats-challenges-and-solution/