

# 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** | **2022-03-15** | **James Roberts** |  |

## 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

James Roberts

## 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?
* Secure communication is the most valuable thing Artemis Financial can offer.
* There is a possibility of international transactions considering users could live

anywhere and investments could be made outside of America.

* The Gramm- Leach-Bliley Act requires Artemis Financial to explain

how the information sharing practices work to the customers to guard

sensitive data.

* Possible external threats are username harvesting, brute force attacks and SQL Injections.
* Important factors that must be considered with the modernization of the website is that open source sometimes grows quickly, so it is important to keep everything updated to maintain the security. If one of the libraries breaks, then the website will also break until the problem is corrected.

## 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: The input validation needs to be checked because users will be asked to input information to log in and complete transactions.
* Cryptography: With personal and financial information being shared online there needs to be steps to keep it safe.
* Code Error: Because the software will be updated often, ensure there is a way to manage the errors that my come up.
* Encapsulation: With all of the data that will be stored, ensuring that the data structures are set up and working correctly.

## 3. Manual Review

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

In the CRUD.java, CRUDController.java, customer.java, DocData, Greeting.java, GreetingController.java, there is no input validation. There needs to be whitelisting, length of input, validating numerical inputs, and open text inputs.

Ensure that the entire website is HTTPS and consistently updated. Ensure URL rewriting, or modifying is not supported. Ensure there is an idle Timeout. Consider using a longer session ID.

Ensure that there is intrusion detection implemented. Also checking for obvious attacks and immutable components. Set up a Tokens Revisited check and a limit on how many attempts can be made to log in from a specific ip address in a set amount of time to prevent polymorphism attacks.

Ensure that there is symmetric and asymmetric keys being used to lock the 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 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

For the dependency bdprov-jdk15on-1.46.jar there are 14 vulnerability codes.

CVE-2013-1624, Allows remote attackers to conduct distinguishing attacks and plain text recovery attacks via statistical analysis of timing data for crafted packets. CVE-2015-7940, Allows does not validate a point within the elliptic curve, making it easier for a remote attacker to obtain a private key. CVE-2016-1000338, Allows no signature on verification creating a possible to inject extra elements in the signature and keep it valid. CVE-2016-1000339,Allows key tacking to happen. CVE-2016-1000341, Allows timing attack. CVE-2016-1000342, Allows injection in signature and still be valid. CVE-2016-1000345, Allows attackers to observe timing and identify when the decryption is failing due to padding. CVE-2016-1000346, Allows the public key is not fully validated. CVE-2016-1000352, This is unsafe and support has been removed. CVE-2017-13098, Allows an attacker to recover a private key from an application. CVE-2018-1000613, allows deserialization of private keys then a new key can be made. CVE-2020-1552, allows a timing issue with the math library that can expose information about the private key. An upgrade of the bouncy castle dependency will fix all of the problems above.

CVE-2016-1000343, Allows up to 1024 bit key size. To fix this make sure that there are parameters on the key pair generator. All of these problems have been found over the last seven years and fixed there is also a lot of documented problems at oracle.com and access.redhat.com.

For the dependency jackson-datablind-2.10.2.jar. There is 1 vulnerability code. CVE-2020-25649, A flaw found in the data bind did not have entity expansion secured properly. This allows vulnerability to xml attacks which can cause data integrity. An upgrade will fix this problem. Oracle.com has a list affected products and versions and patches available.

For the dependency log4j-api-2.12.1.jar. There is 1 vulnerability code. CVE-2020-9488, improper validation of certificate. Allows an SMTPS connection to be intercepted by a man in the middle attack. An upgrade will fix this problem. Oracle.com has a list affected products and versions and patches available.

For the dependency logback-core-1.2.3.jar. There is 1 vulnerability code. CVE-2021-42550. This allows attackers with the required privileges to edit configuration files to create malicious configurations. This is a low level threat and most likely fixed once the log4j-api-2.12.1.jar. Dependency is fixed. There is a full description at jira.qos.ch

For the dependency snakeyaml-1.25.jar. There is 1 vulnerability code. CVE-2017-18640, allows entity expansion during a load operation or a billion laugh attack. An upgrade will fix this problem. Oracle.com has a list affected products and versions and patches available.

For the dependency spring-aop-5.2.3.RELEASE.jar. There are 4 vulnerability codes. CVE-2020-5421 allows RFD attacks is bypasses depending on the browser being used. CVE-2021-22060 allows a user to provide malicious input to cause insertion of additional log entries. CVE-2021-22096 same as the last one. CVE-2021-22118, allows an attack by recreating temporary storage directory. All that is needed to fix this is an upgrade to the latest version. Documentation on tanzu.vmware.com

For the dependency spring-core-5.2.3.RELEASE.jar. There are 4 vulnerability codes. CVE-2020-5421 allows RFD attacks is bypasses depending on the browser being used. CVE-2021-22060 allows a user to provide malicious input to cause insertion of additional log entries. CVE-2021-22096 same as the last one. CVE-2021-22118, allows an attack by recreating temporary storage directory. All that is needed to fix this is an upgrade to the latest version. Documentation on tanzu.vmware.com

For the dependency tomcat-embed-core-9.0.30.jar. There are 15 vulnerability codes.

CVE-2019-17569 This allows for HTTP smuggling. CVE-2020-11996 This allows the HTTP connection to become available to an attacker. CVE-2020-13934 This allows for a out of memory exception leading to a denial of service. CVE-2020-13935This allows for HTTP smuggling. CVE-2020-17527 Allows to an error and closure and information leak between request. CVE-2020-1935This allows for HTTP smuggling. CVE-2020-1938 This allows the HTTP connection to become available to an attacker.

CVE-2020-9484This is from a sufficiently lax filter to allow the attacker provided object to be deserialized; and d) the attacker knows the relative file path from the storage location used by File Store to the file the attacker has control over. CVE-2021-24122 Allows attackers to see source code in some configurations. CVE-2021-25122 Allows duplicate request headers and a limited amount of request body from one request to another meaning user A and user B could both see the results of user A's request.

CVE-2021-25329 This is from a sufficiently lax filter to allow the attacker provided object to be deserialized; and d) the attacker knows the relative file path from the storage location used by File Store to the file the attacker has control over

CVE-2021-30640 allows an attacker to authenticate using variations of a valid user name and/or to bypass some of the protection provided by the Lock Out Realm

CVE-2021-33037 does not correctly parse the HTTP transfer-encoding request header in some circumstances leading to the possibility to request smuggling when used with a reverse proxy. CVE-2021-41079 did not properly validate incoming TLS packets,a specially crafted packet could be used to trigger an infinite loop resulting in a denial of service. CVE-2021-42340 This created a memory leak that, over time, could lead to a denial of service via an Out Of Memory Error. All of the documentation and fixes are on list.apache.org. Once this is updated to the latest version this will all be fixed.

For the dependency tomcat-embed-websocket-9.0.30.jar. There are 17 vulnerability codes.

CVE-2019-17569 This allows for HTTP smuggling. CVE-2020-11996 This allows the HTTP connection to become available to an attacker. CVE-2020-13934 This allows for a out of memory exception leading to a denial of service. CVE-2020-13935This allows for HTTP smuggling. CVE-2020-17527 Allows to an error and closure and information leak between request. CVE-2020-1935This allows for HTTP smuggling. CVE-2020-1938 This allows the HTTP connection to become available to an attacker.

CVE-2020-8022 A Incorrect Default Permissions vulnerability in the packaging of tomcatllows local attackers to escalate from group tomcat to root.

CVE-2020-9484This is from a sufficiently lax filter to allow the attacker provided object to be deserialized; and d) the attacker knows the relative file path from the storage location used by File Store to the file the attacker has control over. CVE-2021-24122 Allows attackers to see source code in some configurations. CVE-2021-25122 Allows duplicate request headers and a limited amount of request body from one request to another meaning user A and user B could both see the results of user A's request.

CVE-2021-25329 This is from a sufficiently lax filter to allow the attacker provided object to be deserialized; and d) the attacker knows the relative file path from the storage location used by File Store to the file the attacker has control over

CVE-2021-30640 allows an attacker to authenticate using variations of a valid user name and/or to bypass some of the protection provided by the Lock Out Realm

CVE-2021-33037 does not correctly parse the HTTP transfer-encoding request header in some circumstances leading to the possibility to request smuggling when used with a reverse proxy. CVE-2021-41079 did not properly validate incoming TLS packets,a specially crafted packet could be used to trigger an infinite loop resulting in a denial of service. CVE-2021-42340 This created a memory leak that, over time, could lead to a denial of service via an Out Of Memory Error. All of the documentation and fixes are on list.apache.org. Once this is updated to the latest version this will all be fixed.

## 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.

After the review there are a few things that need to be reviewed. Ensure that there are input validations on any inputs to limit the chances of an sql injection. Make sure that the entire website is using HTTPS and is consistently updated. Ensure that there is an intrusion detection system implemented. Make sure that there is a limit on the number of log in attempts from a specific ip address in a set amount of time. The last, but most important thing, is to make sure that every dependency is updated when available.