****

# 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** | **1/19/2022** | **Cody Poley** | **Interpreting clients needs drafted** |
| **1.1** | **1/20/2022** | **Cody Poley** | Areas of Security drafted |
| **1.2** | **1/21/2022** | **Cody Poley** | Manual Review drafted |
| **1.3** | **1/22/2022** | **Cody Poley** | Static Testing drafted |
| **1.4** | **1/23/2022** | **Cody Poley** | Mitigation Plan drafted |

## 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

Cody Poley

## 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?

Artemis Financial wants the most current and update software security to protect their web-based software application. The company handles financial information of other clients and needs the API to be well protected from outside sources due to this the value of this security communications are a high priority. The company deals with investments there for the company could deal with international transactions. When dealing with secure communications governmental restrictions should be taken into consideration this information can be found under the Data Security Laws for each state, the Privacy Act of 1974, Electronic Communications Privacy Act, Computer Fraud and Abuse Act, and Cyber Intelligence Sharing and Protection Act. External threats that should be considered are denial of service, distributed denial of service, SQL injection, XSS attacks, hijacking sessions, and zero-day attacks. Modernization requirements that need to be considered are not to use open-source libraries but use the libraries for the language being used in this case Java library. Another consideration that should be considered is evolving web application technologies. Evolving web application technologies should be implement as soon as possible after checking to make sure the software will not make it easier for attacks from different threats.

## 2. Areas of Security

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

* Input Validation should be used to ensure that only data that the application deems good will be used and any other data will be rejected. This will protect the system from attacks through input to the application.
* API security should be used to protect the application from hash based XSS attacks. This will protect the application from malicious code being injected.
* Cryptography should be used to encrypting passwords and ongoing data this will protect the accounts of the clients and protect the information.
* Code error handling should be used secured to ensure that hackers will not be able to tell what the error was this can be done by using error code numbers rather than telling the user what error it is.

## 3. Manual Review

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

* Model is Greeting.java and CRUD.java both the id, content, content1, and content2 are not encrypted, no locks or synchronizations where used and memory was not wiped.
* Controller is GreetingController.java and CRUDController.java, the % is used in String template on the GreetingController.java. The following are not used in the controller error checking, locking or synchronization, and memory clearing.
* Data access object is the customer.java should be encrypted to protect the customers’ accounts numbers and account balance. The values need error handling in case an integer is not enter and range need to be check as well.
* Plug-ins need to be protected from outside sources.

## 4. Static Testing

Run a dependency check on Artemis Financials’ 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

* Bcprov-jdk15on-1.46.jar means in the Bouncy Castle JCE Provider ECIES implementation that allows ECB mode. The solution to fix this problem is updating to the latest Bouncy Castle JCE Provider.
* Hibernate-validator-6.0.18.Final.jar means a flaw in the Hibernate validator causing an invalid EL expressions to be seen as valid. The solution to fix the problem is updating the OneDev to 4.0.3 version or newer.
* jackson-databind-2.10.2.jar means a flaw in the FasterXML Jackson Databind causing weakness that allows XML external entity attacks. The solution to fix the problem is updating the Etherpad, XStream, and TensorFlow.
* Log4j-api-2.12.1.jar means a flaw in the FasterXML Jackson Databind this flaw causes a weakness allows for XML external entity attacks. The solution to fix the problem is by updating the Apache Log4j2 to the latest version.
* logback-core-1.2.3.jar means a flaw in the logback allowing an attacker load code to the LDAP servers. The solution to fix the problem is by updating the sunrpc module of the GNU C Library to the latest version.
* Snakeyaml-1.25.jar means a flaw in the SnakeYAML Alias feature causing an issue that allows an attack during a load operation.
* Spring-app-5.2.3.RELEASE.jar means a flaw in the Spring Framework allowing RFD attacks. The solution to fix the problem is by updating the jsoup to the latest version.
* Spring-core-5.2.3.RELEASE.jar means a flaw in the Spring Framework allowing for malicious input attacks. The solution to fix the problem is by updating the OpenOlat to the latest version.
* Tomcat-embed-core-9.0.30.jar means a flaw in Apache Tomcat allowing for memory leaks. The solution to fix the problem is by updating the Goobi Viewer Core to the latest version.
* Tomcat-embed-websocket-9.0.30.jar means a flaw in the Apache Tomcat allowing for memory leaks. The solution to fix the problem is by updating the Goobi Viewer Core to the latest version.

## 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 Financials’ software application.

To fix the security vulnerabilities for the Financials’ software application, first all passwords and account information including account numbers should be encrypted making it so that no hackers will be able to read it. locks or synchronizations all parts of the program to ensure that no leaks of information are used to hack the program. Memory should also be cleared after each process is completed to ensure that no memory leaks are found. The range and size of all inputs should be checked for any inconsistencies that may cause errors. Access to the code and plugins should be restricted from outside sources to protect from malicious coding. All comments in the program with information explaining how the program works should be remove especially default database usernames and passwords. Error handling needs added to all parts of the program and error code should not tell the user what the problem is but give an error code that the tech support can look up also if the username or password is incorrect the error should not tell the user which one was incorrect. The Bouncy Castle JCE Provider needs to be updated to the latest version to protect the ECB mode. Hibernate validator needs to be updated to the latest version to protect from invalid EL expressions to be seen as valid. The FasterXML Jackson Databind needs to be updated to the latest version to protect from XML external entity attacks. The logback needs to be updated to the latest version to protect from attackers loading code to the LDAP servers. The SnakeYAML needs to be updated to the latest version to protect from attacks during a load operation. The Spring Framework needs to be updated to the latest version to protect from RFD attacks. The Apache Tomcat needs to be updated to the latest version to protect from memory leaks.