

**CS 305 Project One**

**Artemis Financial Vulnerability Assessment Report**

Document Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Comments** |
| **1.0** | **11/13/2021** | **Jayson Wolfe** | Created and finished the project by filling in the entire template |

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

Jayson Wolfe

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?

There is a very high value in secure communications to the company. Nobody would trust a company that was prone to being targeted and hacked. Having secure communications helps the company image seem like it is safe and secured. It lets the customer feel like they can put their trust in the company. This is espesially true when dealing with peoples financial plans, in this scenario the company in question is a financial company holding peoples life savings and retirement funds. Because of the potential of international stock market trading there are the company will produce international transactions or at least needs the ability to be able to accomodate this feature. There are also quite a few government restrictions to consider as the government is really caution about its money. Some potential external threats that might be present now and during the immediate future would be anyone trying to make a quick buck and knows what the company does. Since lots of people are looking to make easy money one would need to protect all the assets against those potential threats. Some modernization requirements require the company to be fully web based in order to benefit from an open source library.

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.

Referring back onto the VAPFD the areas that are applicable to this software application are API, Cryptography, Client/Server, and Code Quality. The reason API is applicable is because we need to limit how much interaction can happen in case an attacker gets in. By limiting the amount of access you can ensure that potential threats would need multiple login credentials in order to get access to client data. Cryptography is relevent to this application because as the software has valuable information being transmitted across the internet and that needs to be protected. Encryptions help protect that kind of data from getting into the wrong hands mid transit. Client/Server is relevent because as a web application there is communication between both the client and server. Having physical security measures on client side to secure data that gets transmitted from the server and vise versa. Code Quality means that the code is up to safety standards by following safe coding practices. You can save a lot of time by prewriting the code using safe security features.

3. Manual Review

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

There were a couple of vulnerabilities within the CRUDController and the DocData files. The CRUDController file has some exposed data under the "business\_name" value. This is because it is sent as a request parameter so it end up leaving this value exposed. Where the DocData files vulnerabilities were in the password reccomendations the passwords were too easy or lax. They need to make more complex passwords so that nobody can have a chance at guessing what the actual password is. Making sure to use safe web services such as HTTPS when sharing valuable information.

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:

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

The vulnerablility code of the known vulnerabilites are CVE-2015-6644, CVE-2020-9488, CVE-2020-13935, CVE-2020-25649. The recomended solution for 6644 is to update the application often to make sure that it stays up to date. The recommended solution for 9488 is to update to a newer version. The reccomended solutions for 13935 and 25649 are also both updating the application and making sure that it stays up to date. Some problems that arise from these vulnerabilities are high memory consumption causing crashes, letting log messages to be revealed, and allowing potential threats to be able to find access into the application.

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.

My Mitigation Plan from the results of the manual review and static testing are as follows, creating a stricter requirement on password generation because passwords are the first layer of defense against unauthorized personel the combonations should not be easily guessable or obtainable. This will help reduce the chance of someone guessing the combonation and getting access that way. One way to do this would be the inclusion of special characters such as ! or & or # this can create quite a few more possible password combinations that make it near impossible to guess at random. Reviewing the code after changes are made in order to make sure that you can catch any security errors. Updating outdated software and other vulnerabilities used within the code is also advised in order to prevent the vulnerabilities that were found during static testing.