****

# 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** | **[Date]** | **[Your Name]** |  |

## 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

Kainan Woodard

## 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 communications have a huge value for the company because they allow the company to ensure that people aren’t able to hack into and view the messages sent between the two companies, because if a competitor were to find out how they made their software and how long it would take them to develop the product they, the competitor, could steal secrets and develop their product first. If the person who’s hacking in isn’t a competitor to the company, they would use the hack on communications to intercept information and attempt to find and use various admin usernames and passwords, this would allow them to gain access to payment information and cause serious problems. The company could have international transactions, but they would most likely just be an intranational business. I don’t remember reading about any government restrictions for communications so I wouldn’t think there are. There could be a ton of potential threats, a few threats that might be present in the future are unsafe data and new vulnerabilities appearing that weren’t problems when the software was originally created. Current threats that would be present now are that they still have to deal with issues like SQL injection and other current external user problems. Especially because this is a company that deals with money people will always be trying new ways to steal money. The requirements of modernization that must be considered will just be evolving web application technologies, as a corporation that handles mass finance using other people’s code will be a huge problem because it all will have to be scanned for security problems and risks.

## 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.

The areas of security that are relevant to this application are Cryptography, client/server, code quality, and encapsulation. The reason that cryptography is important for this application is because the software needs to make sure each clients’ data is protected from potential attacks. The reason why client/server is important because they software needs to ensure that the information the client is accessing is from the server and not being changed by the individual clients that are running. The software also needs to make sure that each client that is being run is separate from one another. Code quality is important because checking for common software problems is important to ensure there aren’t any major attacks that will happen to the software, things like injection and upscaling permissions. The final one is encapsulation which involves the security for the data structures involved. The data structures themselves are important because we need to minimize the runtime for each of the users when they are comparing their usernames and passwords to the ones in the system. Securing the structures is important so someone can’t “map” out the database to find out people’s information.

## 3. Manual Review

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

Looking at the code I can see that the code has variable names that the client has direct access to instead of using server based ones. The code also shows a direct link to the server in the DocData file.

## 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

* CVE-2013-1624
  + Bouncy Castle: TLS implementation doesn’t properly consider timing side-channel attacks on a noncompliant MAC check operation. Attackers can conduct distinguishing attacks for crafted packets.
  + This vulnerability was first found in 2012 and one of the solutions found to correct this is to identify the number of bites, compare it with how many bites should be found and deny the attack.
* CVE-2015-6644
  + Bouncy Castle: Enabling a local malicious application to gain access to a users’ private information
  + This vulnerability has been found but is it is a problem with the previous version as there is no official documentation for a solution.
* CVE-2015-7940
  + Bouncy Castle: Java library before 1.51 does not validate that a point is within the curve, so attackers can obtain private keys.
  + Updating the version of oracle fixes this issue
* CVE-2016-1000338
  + Bouncy Castle: It’s possible to inject extra elements into the sequence making up the signature and it will still validate(could cause invisible data)
  + Updating Oracle fixes this issue
* CVE-2016-1000341
  + Bouncy Castle: allows attackers to gain info about a signature’s k value
  + Updating Oracle fixes this issue
* CVE-2016-1000342
  + Bouncy Castle: “invisible data”
  + Update Oracle
* CVE-2016-1000343
  + Bouncy Castle: key pair gen. is not initialized with DSA
  + Update Oracle
* CVE-2020-10693
  + Hibernate Validator flaw: attackers can bypass input sanitation controls that devs have put in place.
  + Update Red Hat
* CVE-2020-25649
  + FasterXML Jackson Databind flaw allows vulnerability to XML attacks
  + Upgrade Red Hat

There are a ton of other errors but a lot of them are solved if the devs update the software and user parameterization

## 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.

The steps that the developers need to do to fix a lot of the security vulnerabilities are to change the variable names to server-side names. They also need to add query parameterization and use a schema to prevent injection attacks. The biggest fix that the developers can do for this project is to update all of the code to the most recent version available.