****

# 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** | **09/15/2022** | **Kyle Raja** | **Assessment report for Artemis Financial** |

## Client



## Developer

Kyle Raja

## Interpreting Client Needs

1. The client mainly wants to feel like the company cares about their needs and values security. They want to feel safe with their business and feel a sense of professionalism through the process. They want to feel that their client information is protected from cyber-attacks. Not only this, but the client also wants to feel that they can count on important requirements to be completed. The way it works is in a cycle. Artemis Financial has important clients. Their clients have requirements that must be fulfilled. Some of those requirements may involve features, added security, or other enhancements to the application. Those developers try to fulfill their requirements which in turn helps Artemis Financial fulfill their clients’ expectations.
2. The company has the added advantage of being accessed across the network which means that they do have the capability of international transactions. This is advantageous however, makes things are more complicated as far as time zones, different types of currencies, customers expectations, and causes the application to likely be used 24/7.
3. This is something that must be addressed, governmental restrictions regarding communications. The fact that the application can be used internationally means that there will be different types of currency and rules. Also, security will vary between countries and government rules regarding security will also vary. It’s important to understand that cyber attacks could come from anywhere, whether in the United States or in another country. Because we are dealing with transactions, monies, bonds, and stocks, different countries will have specific governmental restrictions in place. For example, some countries require multiple forms of ID and other information before allowing communication regarding transactions.
4. The biggest threat that everyone is aware of are cyber-attacks. Cyber attacks are known for extracting/decoding important client financial data such as credit card information. Cyber security combats breaches from these cyber attackers however more and more ways are being developed by these attackers to bypass security measures. Today we have malware, worms, and other viruses that are a constant threat.
5. Open source is a great and continent resource for web-based applications. From these libraries the application can create workable environments to use functionality in a practice environment or environments specifically for customers. There must be security surrounding these libraries. Another thing that should be considered is where the company stores its source code. They could store it in SVN, GIT, or some other type of version control. I know that Git has great security where repositories can be protected.

## Areas of Security

1. Input validation- This is one of the primary areas where security must be considered. Cyber attackers attempt to access client data through inputs to decode the password. Having input validation and security over that input would block cyber attackers. Input from the user determines what the program will do next. For example, a login screen that asks for username and password. Having security layers such as two-way authentication could be a significant advantage for the users data.
2. API – Api are where two or programs relay information to each other and work together. The application is used over the network making it a web application. The application may interact with different types of API’s that the company created for its clients. These APIs must have security set in place. Typically, APIs will have credentials, tokens, that the company has set so that the API environments are protected over the web. There are some authentication web services that can be used to handle security among these APIs. Typically, a client must be set up and referenced from within the application, perhaps an API resources form that the company has.
3. Cryptography- This is an obvious one. Cryptography is typically useful for sensitive data that is communicated via the web. Cryptography goes hand in hand with encryption which is vital towards masking client information so that cyber attackers can not view this data over the internet. At the end of the day, both of these things is about making sure data is secure as its passed through the different web pages, networks, and servers across the internet. What encryption does is go through two processes, encoding and decoding. When a user inputs sensitive information and save it on a web site let’s say for example, Amazon, their data is likely encoded so that cyber attackers can not access things like their card information. This encryption is done in the code and eventually stored in a database. A popular encoding method is masking the information. Then, when the client needs to access this information for example purchasing an order, the encoded data is then decoded through the code through encryption and the webservice validates the information so that the client can purchase the order.
4. Client/Server- This is similar to the examples I mentioned above regarding how a program or better known as the client, interacts with a network or server. The web-based application interacts with one or more servers to pull or store data. Security measures are typically done on the backend when information is being requested however there is plenty of security options that can be implemented on the frontend UI. Servers are typically considered backend, and they must be protected because they are where all the sensitive client data is stored. Firewalls can be put in place between the network, and these severs to add a layer of security. What can also help are security measures done in the program that help verify security before extracting information or storing information on the server. Client/Server is really this kind of interaction and having strong security in between their communications is vital.
5. The last few areas I want to include together are Code error, Code quality, and encapsulation. These three can go hand in hand. Code error checks for exception handling that the user can come across from wrong inputs. It’s a way to make sure the program should behave like it should. Code Quality is about looking for potential areas and making sure the code is as clean and efficient as possible. Encapsulation involves protecting the code so that the user only sees the most important things inside the program.

## Manual Review

After looking through the code I have some suggestions where I saw certain vulnerabilities. In the “Customer.java” file. I saw that the methods that deal with customer information such as the “showInfo” method and the “deposit” method where public. Its good that the account number was a private member, however these methods are dealing with highly sensitive data, and I recommend they be better protected. Especially, the showInfo method, it returns the user account number. Another area I saw that was vulnerable was the “DocData.java” file. In it I saw a connection string that could potentially be accessed by cyber attackers. Connection strings contain highly sensitive information including the username and password of how to access datasources. I think that this should be better protected.

## Static Testing

After running the dependency check on the project, I found some vulnerabilities that I would like to address:

* bcprov-jdk15on-1.46.jar (bouncy castle) CVE-2016-1000342

The severity of this vulnerability is high, so I wanted to make it aware. This particular ticket noticed an issue where extra signatures could be injected as elements and have them still validated in the program. This means, a cyber attacker could hack into the system, create login information, and be validated to access the program. This I considered was a serious problem.

* hibernate-validator-6.0.18.Final.jar (jackson-databind) CVE-2020-25649

This I saw was also a high priority ticket that should be addressed. The issue is that the entity expansion is not secured. This allows hackers to tamper with the companies XML files that have sensitive data information. XML files for a financial company are extremely important therefore this problem must be addressed.

* snakeyaml-1.25.jar (snakeYAML) CVE-2022-38749

This was a more serious threat to the servers and program. The issue is that untrusted YAML files could bypass DOS and cause a crash in the system. This would make things very bad for the clients and business. Recommended that this vulnerability is fixed soon.

## Mitigation Plan

I have outlined a set of actions that we could implement in order to handle the more serious vulnerabilities that I mentioned in the report. Obviously, the highest tickets should first be investigated because they pose the greatest threat towards the application. Security is priority!

* bcprov-jdk15on-1.46.jar (bouncy castle) CVE-2016-1000342

As mentioned, this vulnerability allows false logins to be used in the system to access the program. This means an attacker can create a login and have our servers validate it. There needs to be stricter validation for the logins. What we could start with are having a requirement for logins to be more complex. A set standard of how long, how many letters, numbers, and special characters, needs to be implemented in creating the login for a user. Second, there needs to be a two-way authentication implemented for the user. The user logs in and a text is sent to their phone number on file. This provides an extra layer of security among the application. Our servers also have protection among the databases which store this login information. I recommend installing a firewall for our servers to detect attackers trying to use false login information. Any login that is used that is not recognized by the server, should be flagged to the IT team members.

* hibernate-validator-6.0.18.Final.jar (jackson-databind) CVE-2020-25649

XML files are probably the most important documents a financial company may have. It has all of the client financial data which is why its so crucial to protect these files. There needs to be better protection for the companies XML files. It should be stored on a separate server and drive that should be managed by the IT team. The IT team could grant certain users to access the server and drive using authentication services windows already has. IT should also be notified why unauthorized access is used on the server. We should also find a way where the development team could encrypt these XML files so that attackers will not easily be able to see what is on them.

* snakeyaml-1.25.jar (snakeYAML) CVE-2022-38749

Crashes are something we really want to avoid. A leak in security that would allow attackers to cause a stackoverflow crash could make the clients feel unsafe and feel that the program is unreliable. As with our XML files, we need better protection for our YAML files also. These could also be placed on a separate server and managed by the IT team. The key point here is that these breaches of security are made aware to the IT team before the incident occurs. Therefore, we must have top of the line servers in place and have the very best firewalls. I recommend antivirus software be installed on the company devices as well. Every team member needs to be aware of which type of documents are the most sensitive. A sense of security awareness must be in every team member so they can report to management if they notice a breach in the program. I recommend that we start implementing certificates among the APIs that each team member will need to install. IT can verify that the user has a certificate and better be able to see a breached user who does not have a certificate in place.