****

# 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** | **9/16/2021** | **Christian Drouin** |  |

## Client



## Developer

Christian Drouin

## 1. Interpreting Client Needs

Secure communications are critical for a web application such as Artemis Financial. This consulting company develops financial plans which means that the data being sent back and forth through their website is loaded with highly sensitive data. Additionally, this application is not limited by location which therefor opens their servicing doors to the world. With this is mind we know that some of Artemis Financials data is being transmitted internationally which is only furthering the security needed to protect its user’s data. There is no special governmental restriction that requires consideration from the client. The secure and trusted communication networks act of 2019 should always be observed and practiced as a standard, but Artemis Financial more than likely won’t be involved with any situations that would require a review of this act. When handling financial data there is always a significant risk of attack. The client should always be on the lookout for hackers and ensuring their software is protected. With a massive update to the client’s system, we must carefully consider all variables of a modern web application. In the subject of security is important to understand that many of the secure methods involve using third party frameworks. Many of these frameworks are secure, but it is vital that developers are specific in choice and application. Poorly chosen and mismanaged frameworks can pose many risks to the web-app.

## 2. Areas of Security

All areas of the vulnerability assessment process flow are important areas of security for the client’s software application. Input validation ensures that only authorized users are accessing their authorized data. Secure APIs are an important feature for Artemis Financial because of their involvement with different savings, retirement, investments, and insurance plans. These different plans will involve different insurance companies and their associated links and secure connections. Cryptography needs to be used to ensure that secure communications are achieved between the client and its users. Client/Server security is always a priority to prevent data attacks form hackers and establish secure connections between the server and the users. Secure coding practices and secure error handling will be put in place to eliminate “weak spots” in code susceptible to hackers. Encapsulation will ensure that users of Artemis Financial have access to their data only and prevent leakage of hierarchy code and data.

## 3. Manual Review

The first vulnerability that our client’s application faces in unsecured input validation which allows for attackers to bypass input sanitation controls. This client’s API is also not secure as it is using an outdated version of Bouncy Castle Java allowing for externally controlled input to select classes or code. There is a third part dependency with Bouncy Castle Java being used in this project that possibly allows for timing side-channel attacks. There are a few different areas of the client’s code that have management errors which is a result of insecure error handling and coding practices exposing the application to attacks.

## 4. Static Testing

Here is a running list of all published vulnerabilities for the Artemis Financials’ application:

CVE-2013-1624 Does not properly consider timing side-channel attacks.

CVE-2015-6644 Could enable a local malicious application to gain users private data.

CVE-2015-7940 Allows for a possible invalid curve attack.

CVE-2016-1000338 // CVE-2016-1000342 Improper verification of cryptographic signature.

CVE-2016-1000339 // CVE-2016-1000343 // CVE-2016-1000344 // CVE-2016-1000352 Cryptographic issues

CVE-2016-1000341 // CVE-2016-1000345 Vulnerable signature generation through timing attack.

CVE-2016-1000346 Key management errors.

CVE-2017-13098 Information exposure through discrepancy.

CVE-2018-1000613 // CVE-2020-9484 Deserialization of untrusted data.

CVE-2018-5382 Improper validation of integrity check value.

CVE-2020-26939 Observable differences in behavior to error inputs.

CVE-2020-9488 Improper certificate validation.

CVE-2017-18640 Entity expansion during a load operation.

CVE-2020-25649 Improper restriction of XML external entity reference.

CVE-2019-17569 // CVE-2020-1935 // CVE-2021-33037 Inconsistent interpretation of HTTP requests.

CVE-2020-11996 Risk of server unresponsiveness.

CVE-2020-13934 Improper release of memory.

CVE-2020-13935 Loop with unreachable exit condition.

CVE-2020-13943 Unexpected responses for unexpected resources.

CVE-2020-17527 // CVE-2021-24122 // CVE-2021-25122 Information exposure.

CVE-2020-1938 // CVE-2021-22118 Improper privilege management.

CVE-2020-8022 Incorrect default permissions.

CVE-2021-30640 Improper authentication.

CVE-2020-10693 Improper input validation.

CVE-2020-5421 // CVE-2020-5421 Bypassed RFD protections.

## 5. Mitigation Plan

There are serious issues that need to be addressed with Artemis Financials’ application. Many of the vulnerabilities identified in the previous section of this report involve outdated dependencies. These issues are mostly easy to fix by ensuring that the most recent versions of software are being implemented. There are other issues that don’t fit the scope of an easy update. Issues with input validation, API security, cryptography and code error are of largest concern. These issues can be fixed, but it will take time and thoughtful developers. Using query parameterization, we can eliminate the risks of input and certificate validation. There are many issues this site is hosting that this solution can heal. API security needs to be enhanced with the use of tokens and gateways. Cryptography will need a significant amount of review because there are multiple errors in the current application that relate to this issue. These issues will be dealt with individually and various tactics will be used to enhance these cryptographic methods. Lastly code error is an issue throughout this application that can be mostly fixed. Most of the code error relates to other issues that will be fixed as outlined above.