****

# 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 5](#_Toc32574612)

[3. Manual Review 5](#_Toc32574613)

[4. Static Testing 5](#_Toc32574614)

[5. Mitigation Plan 8](#_Toc32574615)

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **01/27/2024** | **Pooja Singh** |  |

## Client



## Instructions

Submit this completed vulnerability assessment report. Replace the bracketed text with the relevant information. In the report, identify your findings of security vulnerabilities and provide recommendations for the next steps to remedy the issues you have found.

* Respond to the five steps outlined below and include your findings.
* Respond using your own words. You may also choose to include images or supporting materials. If you include them, make certain to insert them in all the relevant locations in the document.
* Refer to the Project One Guidelines and Rubric for more detailed instructions about each section of the template.

## Developer

Pooja Singh

## Interpreting Client Needs

Artemis Financial is a financial planning consulting firm that wants to improve the security of its web application and modernize its business practices. Artemis Financial prioritizes secure communications in order to protect sensitive customer information, such as data on investments, insurance, retirement, and savings. This is because the company focuses on creating personalized financial plans.

1. **Value of Secure Communications:**

* For Artemis Financial, secure communications are essential because they guarantee the privacy and accuracy of client information while also building consumer confidence.
* Data breaches, illegal access, and possible financial losses from stolen information are all prevented by securing communications.

1. **International Transactions:**

* Due to its international operations, Artemis Financial must take strong security precautions to safeguard confidential financial information while it is being transferred between countries.
* Sustaining customer confidence and meeting legal obligations necessitates adherence to global data protection rules.

1. **Governmental Restrictions:**

* Maintaining the integrity of Artemis Financials’ activities and avoiding legal ramifications require strict adherence to federal requirements pertaining to secure communications.
* Artemis Financial maintains its reputation as a reliable entity in the financial consulting field by abiding industry norms and laws.

1. **External Threats:**

* Some of the external threats that Artemis Financial is subject to are as follows:
  + Cyberattacks targeting financial data for monetary gain.
  + Malicious parties trying to get unauthorized access to the web application by taking advantage of flaws in it.
  + Phishing attempts meant to obtain private information from clients or staff members.
* To successfully minimize these threats, proactive security measures and ongoing monitoring are required.

1. **Modernization Requirements:**

* In order to improve the security of its web application, Artemis Financial needs to take into account a number of modernizing requirements, which include:
  + Implementation of updates and patches regularly to address vulnerabilities in open-source libraries that are used within the application.
  + Integration of web application technologies that are evolving to maintain compatibility with industry standards and stay ahead of new threats that emerge.

## Areas of Security

Upon examination of Artemis Financial's web-based application, the following security areas are considered relevant:

* **Authorization and authentication**: Making sure that only people with permission can access vital financial data.
* **Validation and Sanitization**: Validating and cleaning user inputs is a means of preventing injection attacks and data tampering.
* **Session Management:** User sessions must be secured to prevent access from unauthorized parties and to prevent hijacking of sessions.
* **Data encryption:** Using encryption techniques to safeguard sensitive data during transmission and storage.
* **Error Handling and Logging:** To effectively identify and address security events, strong error handling and logging procedures must be implemented.

## Manual Review

Manual code base inspection, in accordance with the Vulnerability Assessment Process Flow Diagram, identified vulnerabilities in particular application architecture sections:

* **Insecure Direct Object References:** User IDs are shown without the necessary authorization approvals. Unauthorized access to private user information may result from this vulnerability.
* **Cross-Site Scripting (XSS) Vulnerabilities:** Insufficient input sanitization can potentially allow attackers to insert malicious code. Client-side attacks could compromise user data or hijack sessions as a result of this vulnerability.
* **Insufficient Password Security:** Passwords are saved in plaintext rather than being hashed for safe storage. User credentials are susceptible to theft and unauthorized access due to this vulnerability.
* **Lack of CSRF Protection:** Certain vital actions are susceptible to Cross-Site Request Forgery (CSRF) attacks because they do not have CSRF tokens. This vulnerability can be used by attackers to carry out illegal transactions on behalf of users who have been granted authentication.

## Static Testing

| **CVE Identifier** | **Description** | **Recommended Solution** | **Attribution** |
| --- | --- | --- | --- |
| CVE-2013-1624 | The TLS implementation in the Bouncy Castle Java library before 1.48 and C# library before 1.8 does not properly consider timing side-channel attacks on a noncompliant MAC check operation during the processing of malformed CBC padding, which allows remote attackers to conduct distinguishing attacks and plaintext-recovery attacks via statistical analysis of timing data for crafted packets, a related issue to CVE-2013-0169. | Upgrade the Bouncy Castle Java library to version 1.48 or later. | [MISC](http://www.isg.rhul.ac.uk/tls/TLStiming.pdf), MLIST |
| CVE-2015-6644 | Bouncy Castle in Android before 5.1.1 LMY49F and 6.0 before 2016-01-01 allows attackers to obtain sensitive information via a crafted application, aka internal bug 24106146. | Update the Bouncy Castle library used in Android to version 5.1.1 LMY49F or later. | Internal bug 24106146 |
| CVE-2015-7940 | The Bouncy Castle Java library before 1.51 does not validate a point is within the elliptic curve, which makes it easier for remote attackers to obtain private keys via a series of crafted elliptic curve Diffie Hellman (ECDH) key exchanges, aka an "invalid curve attack." | Upgrade the Bouncy Castle Java library to version 1.51 or later. | - |
| CVE-2016-1000338 | In Bouncy Castle JCE Provider version 1.55 and earlier, the DSA does not fully validate ASN.1 encoding of the signature on verification. | Update the Bouncy Castle JCE Provider to version 1.56 or later. | lucene-solr-user |
| CVE-2016-1000339 | In the Bouncy Castle JCE Provider version 1.55 and earlier, the primary engine class used for AES was AESFastEngine, which could leak information on the AES key being used. | Upgrade the Bouncy Castle JCE Provider to version 1.56 or later. | - |
| CVE-2016-1000341 | In the Bouncy Castle JCE Provider version 1.55 and earlier, DSA signature generation is vulnerable to a timing attack. | Update the Bouncy Castle JCE Provider to version 1.56 or later. | - |
| CVE-2016-1000342 | In the Bouncy Castle JCE Provider version 1.55 and earlier, ECDSA does not fully validate ASN.1 encoding of the signature on verification. | Update the Bouncy Castle JCE Provider to version 1.56 or later. | - |
| CVE-2016-1000343 | In the Bouncy Castle JCE Provider version 1.55 and earlier, the DSA key pair generator generates a weak private key if used with default values. | Update the Bouncy Castle JCE Provider to version 1.56 or later. | - |
| CVE-2016-1000344 | In the Bouncy Castle JCE Provider version 1.55 and earlier, the DHIES implementation allowed the use of ECB mode, regarded as unsafe, which has been removed from the provider. | Update the Bouncy Castle JCE Provider to version 1.56 or later. | - |
| CVE-2016-1000345 | In the Bouncy Castle JCE Provider version 1.55 and earlier, the DHIES/ECIES CBC mode was vulnerable to a padding oracle attack. | Update the Bouncy Castle JCE Provider to version 1.56 or later. | - |
| CVE-2016-1000346 | In the Bouncy Castle JCE Provider version 1.55 and earlier, the other party DH public key is not fully validated, which can reveal details about the other party's private key where static Diffie-Hellman is in use. | Update the Bouncy Castle JCE Provider to version 1.56 or later. | - |
| CVE-2016-1000352 | In the Bouncy Castle JCE Provider version 1.55 and earlier, the ECIES implementation allowed the use of ECB mode, regarded as unsafe, which has been removed from the provider. | Update the Bouncy Castle JCE Provider to version 1.56 or later. | - |
| CVE-2017-13098 | BouncyCastle TLS prior to version 1.0.3, when configured to use the JCE for cryptographic functions, provides a weak Bleichenbacher oracle when any TLS cipher suite using RSA key exchange is negotiated. | Upgrade to BouncyCastle TLS version 1.0.3 or later. | [MISC](https://robotattack.org/) |
| CVE-2018-5382 | The default BKS keystore uses an HMAC that is only 16 bits long, which can compromise the integrity of a BKS keystore. | Upgrade to Bouncy Castle version 1.47 or later. | BID |
| CVE-2020-0187 | In engineSetMode of BaseBlockCipher.java, there is a possible incorrect cryptographic algorithm chosen due to an incomplete comparison. | Apply the patch provided by the vendor or update to the latest version. | - |
| CVE-2020-26939 | In Legion of the Bouncy Castle BC before 1.61 and BC-FJA before 1.0.1.2, attackers can obtain sensitive information about a private exponent due to Observable Differences in Behavior to Error Inputs. | Upgrade to Bouncy Castle version 1.61 or later. | - |
| CVE-2023-33201 | Bouncy Castle For Java before 1.74 is affected by an LDAP injection vulnerability. | Upgrade to Bouncy Castle version 1.74 or later. | - |

## Mitigation Plan

The manual review and static testing results indicate that the following actions should be taken to address the security vulnerabilities found:

* To mitigate all vulnerabilities found, update the Bouncy Castle library immediately to the recommended version.
* To prevent unwanted access and limit access to important resources, use Role-Based Access Control (RBAC).
* To reduce injection vulnerabilities like SQL injection and XSS, improve input validation techniques and cleanse user inputs.
* To prevent unwanted access, encrypt critical data both in transit and at rest using industry-standard encryption techniques.
* In order to mitigate known vulnerabilities and enhance overall security posture, update libraries and dependencies to the most recent versions.
* To reduce CSRF attacks that target important actions within the application, implement CSRF tokens and enforce CSRF protection.