****

# 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** | **9/13/23** | **John Barden** | **Started code security evaluation** |

## Client



## Developer

John Barden

## Interpreting Client Needs

The company being assisted is Artemis Financial, they handle there customers funds and provide financial storage and guidance with there financial future. Due to the nature of the business the company is responsible for sending and storing important customer information. This information can allow restricted access to accounts and their funds and the information to steal their identity. The company needs to keep this information secured. Artemis Financial description does not implicitly state the company is a global partner. Regardless if the company makes international transactions, attackers will attack any vulnerability.   
  
Gramm-Leach-Bliley Act applies to financial institutions, there are Security and Privacy rules required by the institution. The act provides the protection of nonpublic information and provide privacy policies. The client needs to be informed on what information the company collects. This needs to be sent annually. The institution is required to secure client information and protect against unauthorized access and use of accounts and its information. The information needs multiple data security including encryption, access rights and protect against data breeches. Any government agencies responsible for enforcing GLBA can conduct investigations to verify the security of the information.   
  
With ever changing technology and code refactoring to implement new features can develop holes. Making sure that everything is up to date will help alleviate majority of any current or future issues as open source libraries security issues are often addressed in updates. Using an API creates a endpoint to exploit with a denial of service attack a common attack. Malware and Ransomware are a internal attack often used in combination with a phishing attack to gain access and cause chaos or encrypt the system and requires money for the key (Ransomware). Any external physical attack are always always present and need to be address by the institution to inform there employees. Phishing campaigns through emails that sends them to a phishing course if a unsecured link is pressed. The course should be required with a interactive complete to verify course completion. This can be modified later to address any changes to phishing attacks.   
  
Open-source libraries provide quick code development and usually very secure as many review and use the code. Any issues the community find are addressed by the developer and patches are sent out. If these updates are not implemented the now known security vulnerability is announced to all the “Black hat” hackers to figure out how to exploit it further. Hole and patches will continue to occur throughout the codes lifetime and even if it is refactored. Cyber security is ever evolving and new exploits or attacks are created by the evolution difference between the code.

## Areas of Security

The main areas of security concerns for this software are.

* APIs: Verifying the API sends and receives information within standards and can handle DdoS attack be managing server load. This is the public connection to the software so the API is the most important to be secure.
* Cryptography: Verifying the information is sufficiently encrypted to provide time for the information to be deciphered in a event a data breach does occur.
* Encapsulation: Using correctly setup methods and members to restrict the chance of information modification. This protects the objects member variables which could be the users name, address, banking information, ect.

Reviewing these three areas of attack will provide the greatest general increase to the companies internet security. APIs is where the information will be extracted from so that is the focus of the security plan, next would be secure Artemis Financials client information. By encrypting the information before being sent out will mitigate data breech damage, while encapsulation will prevent the information within the object from being modified.

## Manual Review

The CRUD.java file has correct encapsulation having both content and content2 private final prevents any changes to the object after creation. Only way to modify the object is to destroy and recreate it, with this creation style we can use checksum to verify it the information has been modified when sending or receiving the information. CRUDController.java imports from a springframework library, need to verify if the library has any security issues and update if there are vulnerabilities. The API call RequestMapping is a read command, there seems to be no error handling. If a black hat hacker figures out the variable storing the business name they can inject SQL injection to steal information from the server. The customer.java file the account\_balance should be private to hidden the member, public should only be used if needed, default should be private. There should be another verification before the account is modified. DocData.java has error checking and handling around the connecting protocol. Greeting.java handles greeting the user with their information. GreetingController.java greets the user using a API call, it uses a greeting message from a greeting class. There is default value so error handling is handled. MyDateTime.java is not fully implemented with empty methods. This is a major security vulnerability as it will allow an attack access to a empty class method to run code or enter a infinite loop, this is also connected to the restservice package which could allows hackers to manipulate the API.

## Static Testing

|  |  |
| --- | --- |
| bcprov-jdk15on-1.46.jar | There are 13 CVE with high security vulnerability. The most recent CVE code is from 07/04/2023 that effects Java before 1.74. (**CVE-2023-33201)**  Needs update.  NATIONAL VULNERABILITY DATABASE, W. (2023). *You are viewing this page in an unauthorized frame window.* NVD. https://nvd.nist.gov/vuln/detail/CVE-2023-33201 |
| hibernate-validator-6.0.18.Final.jar | Has CVE-2020-10683 has a input sanitation bug that bypasses sanitation checks. This was resolved in the 6.0.20 version, current version is 6.0.18.  Needs update. |
| jackson-databind-2.10.2.jar | Has 6 CVE vulnerabilities, current version 2.10.2. Most recent CVE: CVE-2022-42004 was resolved in 2.12.7.1. There is a disputed CVE: CVE-2023-35116 that should be reviewed. FasterXML, L. (2023). *FasterXML/jackson-databind: General Data-binding package for Jackson (2.x): Works on streaming API (CORE) implementation(s)*. GitHub. https://github.com/FasterXML/jackson-databind |
| log4j-api-2.12.1.jar | Has a certificate validation issue that was resolved in 2.12.3, current version 2.12.1.  Needs update. |
| logback-core-1.2.3.jar | Has a vulnerability to allow malicious code. Issue was resolved in 1.2.7, current version is 1.2.3.  Needs update. |
| snakeyaml-1.25.jar | Has 8 CVE issues, running version 1.25. Most recent CVE: CVE-2022-1471 that was resolved in version 2.0.  Needs update. |
| spring-boot-2.2.4.RELEASE.jar | Has 3 CVE issues, running version 2.2.4. Most recent CVE: CVE-20230-20873 was resolved in 2.5.15.   Needs update. |
| spring-boot-starter-web-2.2.4.RELEASE.jar | Has 3 CVE issues, running version 2.2.4. Most recent CVE: CVE-20230-20873 was resolved in 2.5.15.   Needs update. |
| spring-core-5.2.3.RELEASE.jar | Has 11 CVE issues, running version 5.2.3. CVE-2022-22965 has a CISA code injection warning. This vulnerability is excluded in 5.20.   CVE-2023-20863 is the most recent CVE. Issue resolved in 5.2.24.   CVE-2020-5421 has a RFD attack from CVE-2015-5211. This was resolved after 5.2.0 and versions up to (excluding) 5.2.9. |
| spring-web-5.2.3.RELEASE.jar | Has 12 CVE issues, running version 5.2.3. CVE-2022-22965 has a CISA code injection warning. This vulnerability is excluded in 5.20.  The most recent CVE: CVE-2016-1000027 resolved in 6.0.0.  Needs update. |
| spring-webmvc-5.2.3.RELEASE.jar | Has 11 CVE issues, running version 5.2.3. CVE-2022-22965 has a CISA code injection warning. This vulnerability is excluded in 5.20.  Most recent CVE: CVE-2020-5421 issue present after 5.2.0, and resolved in 5.2.9. |
| tomcat-embed-core-9.0.30.jar | Has 21 CVE issues, running version 9.0.30. CVE-2020-1938 has a CISA where a regular HTTP can have higher trust then expected. Need to upgrade to version 9.0.31, 8.5.51, or 7.0.100 which will requires small adjustments.  Most recent CVE: CVE-2023-41080 issue was resolved in version after 9.0.79.  Needs update and modification. |
| tomcat-embed-websocket-9.0.30.jar | Has 22 CVE issues, running version 9.0.30. CVE-2020-1938 has a CISA where a regular HTTP can have higher trust then expected. Need to upgrade to version 9.0.31, 8.5.51, or 7.0.100 which will requires small adjustments.  Most recent CVE: CVE-2023-41080 issue was resolved in version after 9.0.79.  Needs update and modification. |

## Mitigation Plan

The majority of the dependencies require updates, there are a couple coding modification that can be done to further improve the systems security. Addressing the issues stated within the manual review will improve the code handling and restrict manipulation of the client information. Error checking will help prevent unexpected actions within the code.