**Module Three Project**

Nathalie Morales

CS-305: Software Security

Southern New Hampshire University

05/28/2024

## Ensuring Secure Communication and Data Integrity at Artemis Financial

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **05/28/2024** | **Nathalie Morales** |  |

## Client



## Developer

Student Name

**1. Interpreting Client Needs**

Secure communication is primarily implemented in real-time applications to protect the data transfer link between the client and the server so that there cannot be any violation of data. Artemis Financial plans and manages customers' funds; therefore, protecting this information from unauthorized individuals is paramount. Hence, the use of secure communication should be of paramount importance in any interaction in which clients are involved.

Artemis Financial also faces significant restrictions because it is a financial institution that cannot violate government legislation for transactions and communications. Some of the risks may include people looking for the personal identity details of clients or any related organization. One of the significant drawbacks is that if the security of the API is not well dealt with, there could be information leakage (Artemis Takes Security Seriously—Here’s How We Do It., n.d.). To hedge such risks, the company should demand two-factor authentication to avoid unauthorized access to the accounts, and all correspondence should happen through the HTTPS level.

**2. Areas of Security**

Writing secure code is an essential practice concerning creating well-structured code that increases safety and protects from any outside intervention to benefit the company and the client. Correcting codes poses significant risks because they determine the stability and soundness of the system.

APIs are especially important in this regard given that the Web services utilize primarily RESTful APIs, which must be secure to ensure the data is safeguarded during the transmission. Furthermore, this branch is essential because every time the RESTful API receives data input from the user, it is essential to cleanse and sanitize this input to eliminate the possibility of injection attacks. Measures implemented in secure coding help consider all these factors, thus affording a firm and dependable approach to protecting users’ privacy and valuable data.

**3. Manual Review**

1. The fact that the Service does not utilize HTTPS is that it is advised to use it when passing on sensitive information
2. No authentication system is in use for verification purposes.
3. A wrong input is accepted without validation, meaning outsiders can quickly enter the system.
4. In the CRUD Controller class, business name parameters are provided in the request. This can still be considered vulnerable because it can leak information to outsiders.

**4. Static Testing**

bcprov-jdk15on-1.46.jar – several vulnerabilities on 1.46; update to the latest version.

CVE-2013-1624

CVE-2015-6644

CVE-2015-7940

CVE-2016-1000338

CVE-2016-1000339

CVE-2016-1000341

CVE-2016-1000342

CVE-2016-1000343

CVE-2016-1000344

CVE-2016-1000345

CVE-2016-1000346

CVE-2016-1000352

CVE-2017-13098

CVE-2018-1000613

CVE-2018-5382

Log4j-api-2.12.1.jar – one vulnerability, update to the latest version.

CVE-2020-9488

Snakeyalm-1.25.jar – one vulnerability; update to the latest version.

CVE-2017-18640

Jackson-data-bind-2.10.2.jar – one vulnerability, update to the latest version.

CVE-2020-25649

Tomcat-embed-core-9.0.30.jar – several vulnerabilities, update to latest tomcat version.

CVE-2019-17569

CVE-2020-11996

CVE-2020-13934

CVE-2020-13935

CVE-2020-13943

CVE-2020-17527

CVE-2020-1935

CVE-2020-1938

CVE-2020-8022

CVE-2020-9484

CVE-2021-24122

Hibernate-validator-6.0.18.Final.jar – one vulnerability, update to the latest version.

CVE-202-10693

Spring-core-5.2.3.RELEASE.jar – one vulnerability, update to the latest version.

CVE-2020-5421

**5. Mitigation Plan**

To tackle current and future problems, several measures are suggested to be followed to secure company and client data. First, we should change all communications to HTTPS to minimize the chances of unauthorized intrusions. Second, request parameters should be moved to headers to provide more security. Third, all the information about business names in the hard-coded database credentials should be masked. Fourth, there is a need to enable two-factored authentication systems since this would help protect users’ information. Lastly, we will upgrade everything depending on the result of the latest dependency checks to ensure safety and stability.

**References**

Artemis Takes Security Seriously—Here’s How We Do It. (n.d.). Www.artemishealth.com. Retrieved May 24, 2024, from <https://www.artemishealth.com/blog/how-artemis-health-keeps-your-benefits-data-secure>