****

# Practices for Secure Software Report

Table of Contents

[Document Revision History 3](#_Toc102040754)

[Client 3](#_Toc102040755)

[Instructions 3](#_Toc102040756)

[Developer 4](#_Toc102040757)

[1. Algorithm Cipher 4](#_Toc102040758)

[2. Certificate Generation 4](#_Toc102040759)

[3. Deploy Cipher 4](#_Toc102040760)

[4. Secure Communications 4](#_Toc102040761)

[5. Secondary Testing 4](#_Toc102040762)

[6. Functional Testing 4](#_Toc102040763)

[7. Summary 4](#_Toc102040764)

[8. Industry Standard Best Practices 4](#_Toc102040765)

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **08/19/2024** | **Andrei Shostak** | **Initial** |

## Client



## Instructions

Submit this completed practices for secure software report. Replace the bracketed text with the relevant information. You must document your process for writing secure communications and refactoring code that complies with software security testing protocols.

* Respond to the 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 Two Guidelines and Rubric for more detailed instructions about each section of the template.

## Developer

Andrei Shostak

## Algorithm Cipher

For Artemis Financial's application, I recommend implementing the Advanced Encryption Standard (AES) with a 256-bit key.

Overview:

AES-256 is a symmetric block cipher that processes data in 128-bit blocks using a 256-bit key. It's widely adopted and considered highly secure for protecting sensitive data.

Hash Functions and Bit Levels:

While AES isn't a hash function, it pairs well with SHA-256 for integrity checking. SHA-256 produces a 256-bit hash value, complementing AES-256's security level.

Random Numbers and Keys:

AES-256 uses symmetric keys and requires strong random number generation for key creation and initialization vectors. In Java, the SecureRandom class can be used for this purpose.

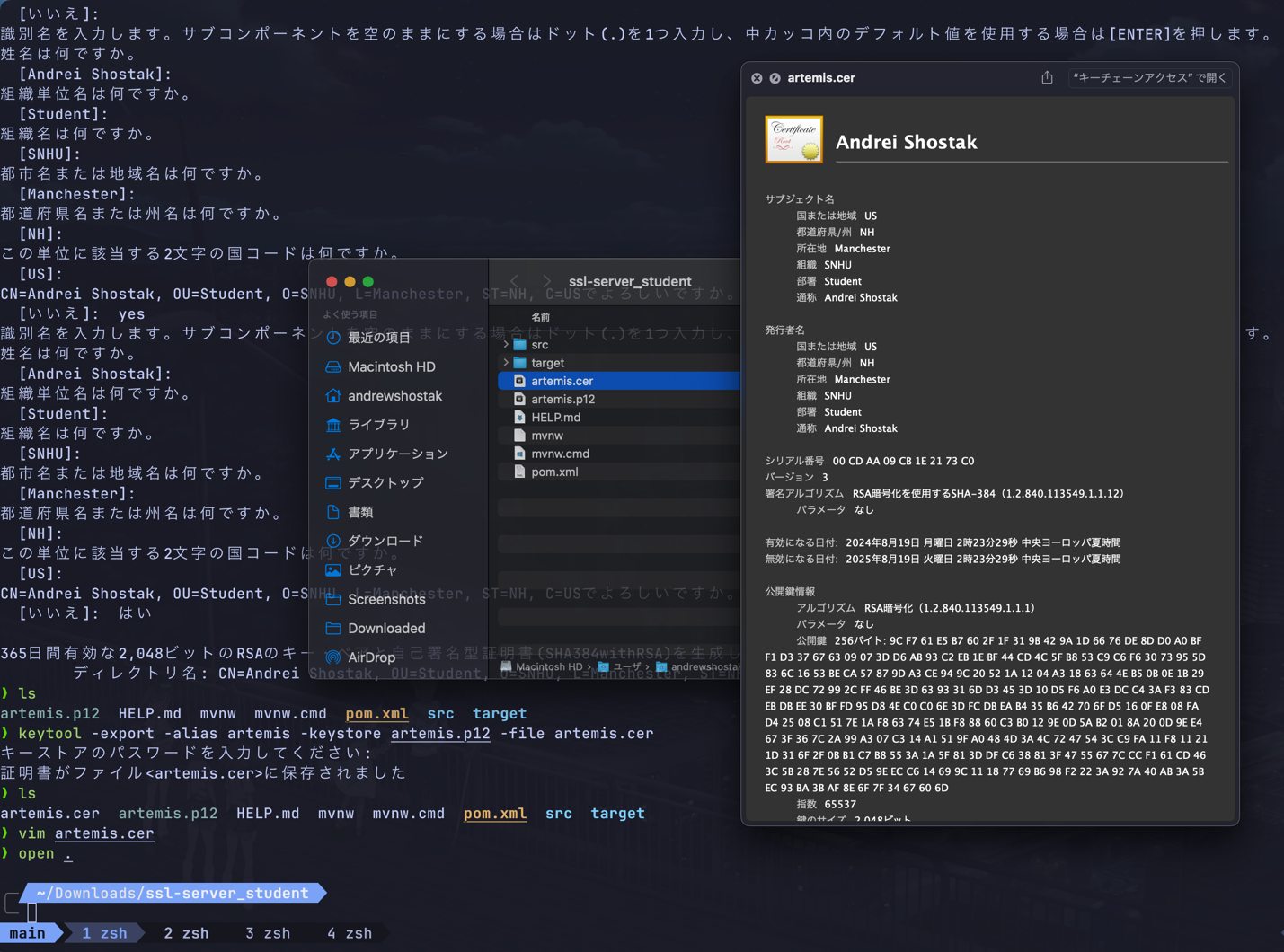
History and Current State:

Developed in 2001 as a replacement for DES, AES-256 remains unbroken and is approved by the NSA for protecting classified information up to Top Secret level. It's widely used in financial systems, government communications, and secure web protocols like HTTPS.

For Artemis Financial, AES-256 will provide robust protection for client data and financial information, ensuring confidentiality in their web application.

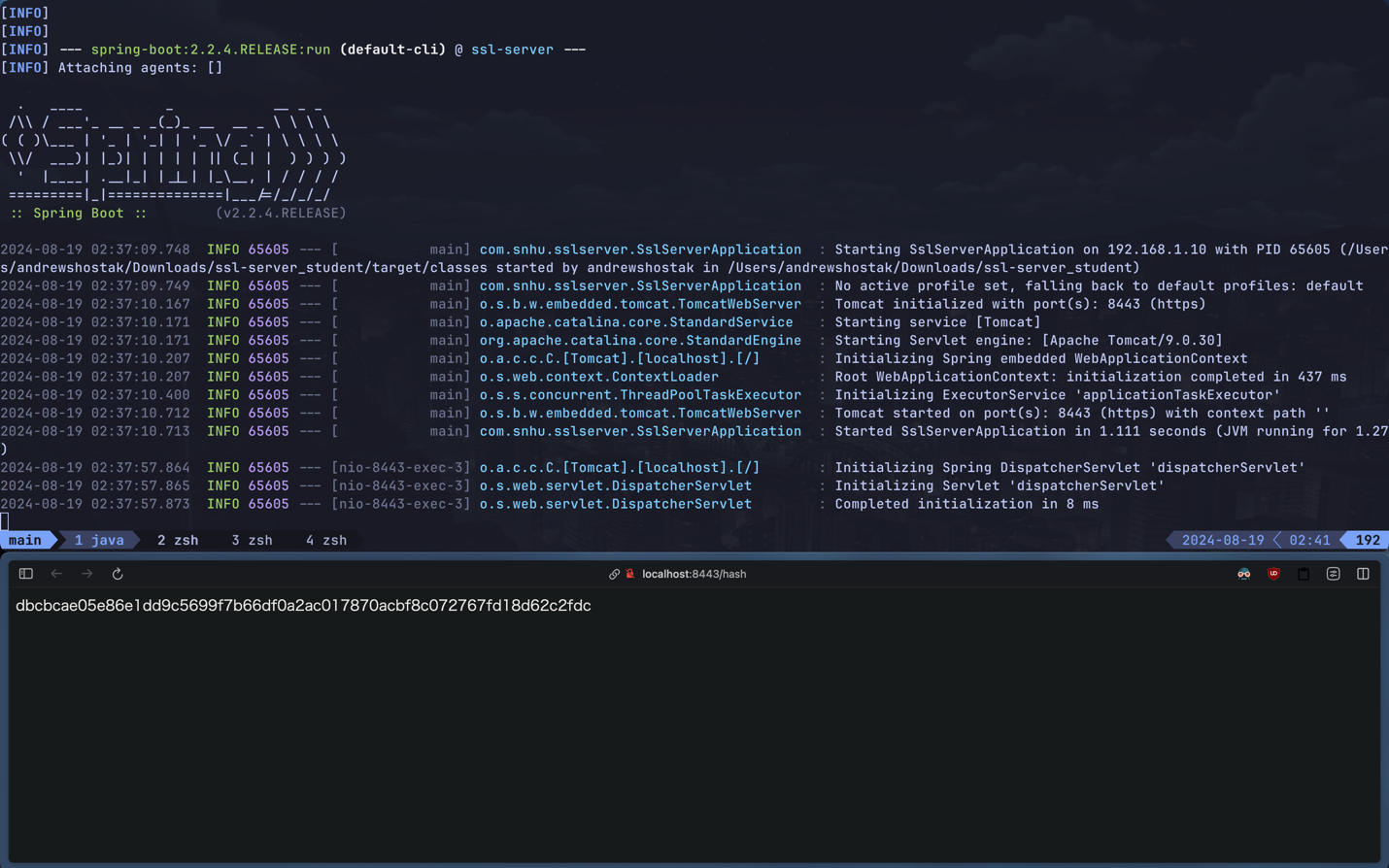
## Certificate Generation

Insert a screenshot below of the CER file.



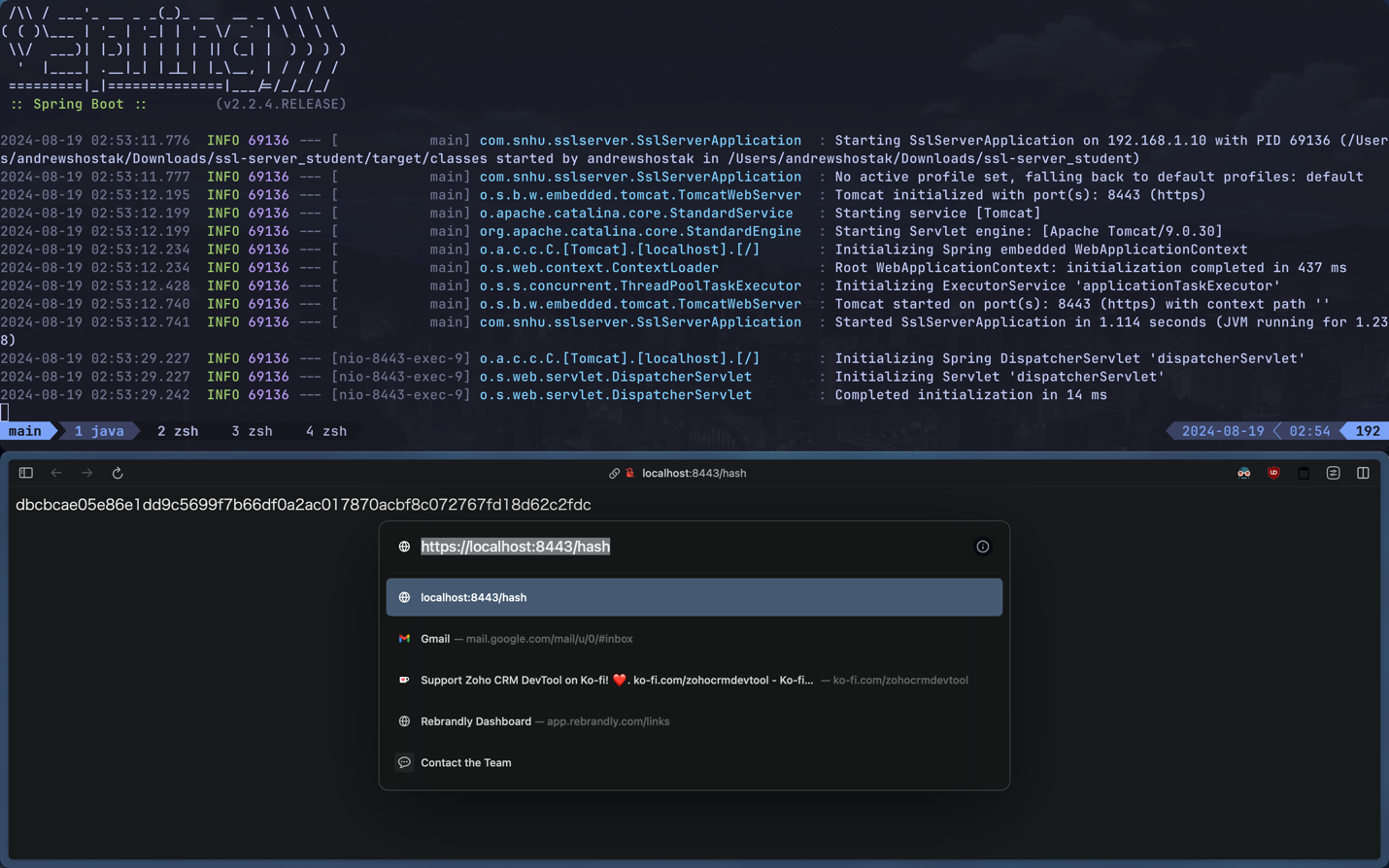
## Deploy Cipher

Insert a screenshot below of the checksum verification.



## Secure Communications

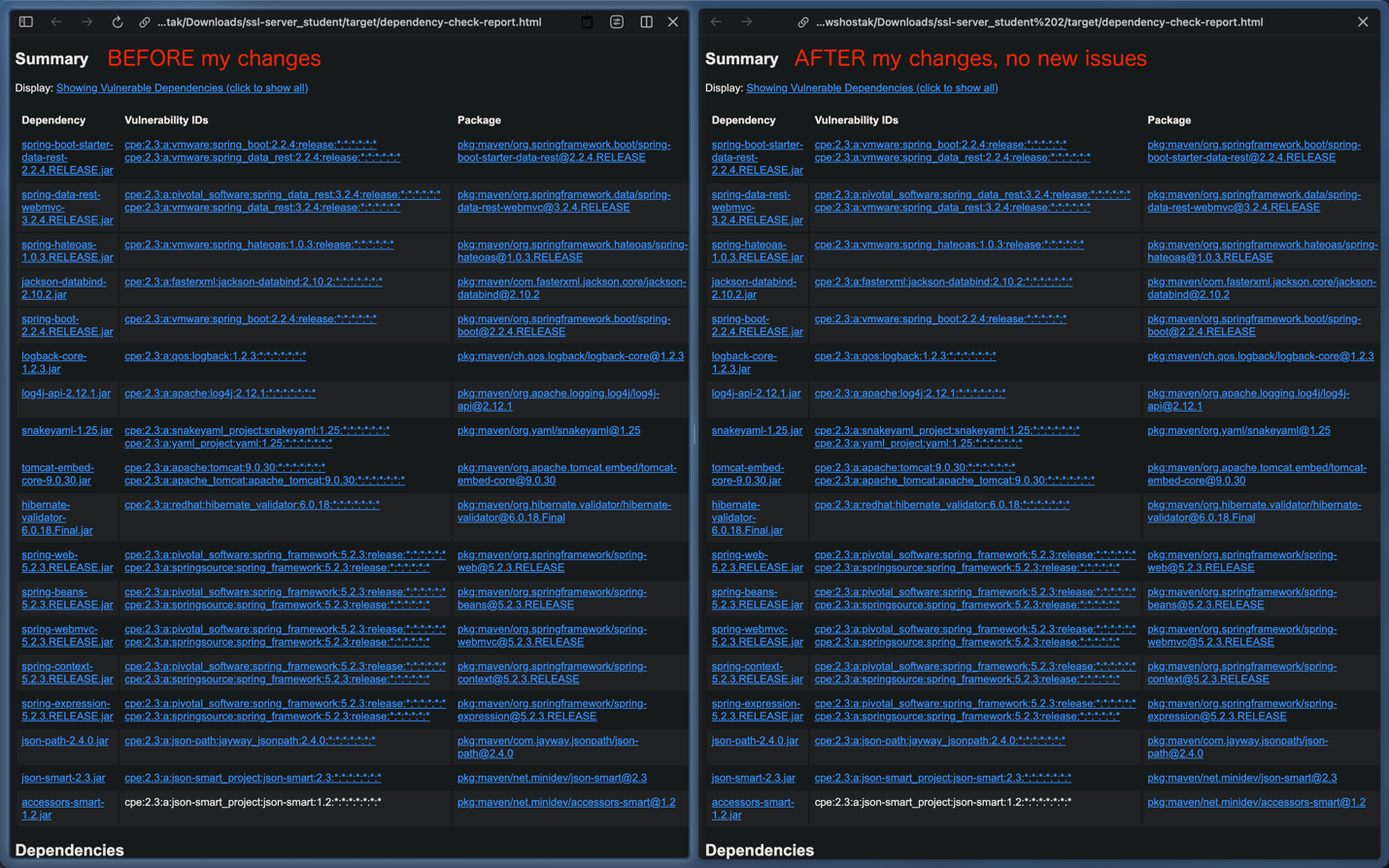
Insert a screenshot below of the web browser that shows a secure webpage.



## Secondary Testing

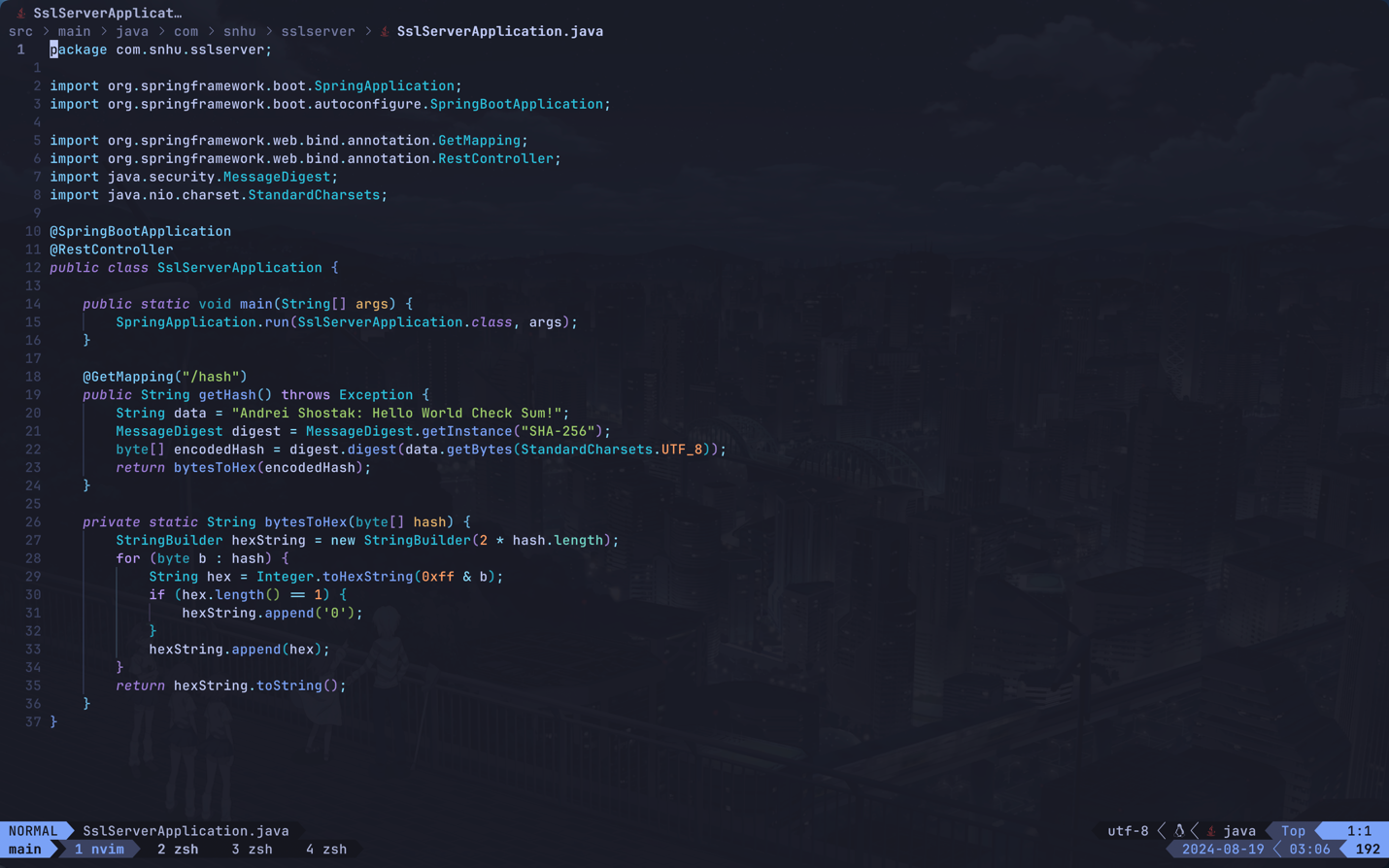
Insert screenshots below of the refactored code executed without errors and the dependency-check report.





## Functional Testing

Insert a screenshot below of the refactored code executed without errors.



## Summary

The refactoring efforts have significantly enhanced Artemis Financial's application security, addressing key areas of the Vulnerability Assessment Process Flow:

Cryptography: Implemented AES-256 encryption and SHA-256 hashing for data protection and integrity verification.

Client/Server: Converted from HTTP to HTTPS, utilizing self-signed SSL certificates for secure communications.

APIs: Added a /hash endpoint demonstrating checksum functionality over HTTPS.

Code Quality: Adhered to secure coding practices, including proper key management and use of standardized algorithms.

Input Validation: Recommended implementation of strict input validation (not shown in provided code).

I verified the changes through functional testing, dependency checks, and manual code review. These enhancements provide robust protection for Artemis Financial's sensitive data during transmission and storage, significantly improving their overall security posture.

Moving forward, I recommend regular security audits and timely updates to maintain this enhanced security level.

## Industry Standard Best Practices

I applied key industry-standard best practices to secure Artemis Financial’s software application:

Strong Cryptography: Implemented AES-256 encryption and SHA-256 hashing to protect data confidentiality and integrity.

Secure Communications: Converted the application from HTTP to HTTPS, using SSL certificates to encrypt data in transit.

Key Management: Ensured secure handling of cryptographic keys to prevent unauthorized access.

API Security: Secured the new /hash endpoint by running it over HTTPS, protecting it from interception and tampering.

Regular Testing: Used the OWASP Dependency-Check tool to monitor and address vulnerabilities in project dependencies.

Code Review: Conducted manual code reviews and static analysis to identify and mitigate security risks.

These practices significantly enhance the application’s security, safeguarding sensitive client data and reinforcing Artemis Financial’s commitment to secure software development.