****

# 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** | **2/23/25** | **Aeriel Denmark** | **Final Submission** |

## Client



Artemis Financial is a financial consulting company that provides customers with individualized financial plans, including savings, retirement, investments, and insurance. The company wants to modernize its operations and enhance its software security by implementing a file verification step to ensure secure communications when transferring data.

## Developer

Aeriel Denmark

## Algorithm Cipher

To enhance security for Artemis Financial’s application, I implemented **SHA-256** as the cryptographic hash function. SHA-256 is a strong, widely used hashing algorithm that provides a secure way to generate unique data fingerprints.

* **Hash Function & Bit Level:** SHA-256 produces a 256-bit hash value, ensuring security against brute-force attacks.
* **Random Numbers & Keys:** Hashing differs from encryption because it does not require keys; instead, it generates fixed-length output from variable-length input. This ensures data integrity rather than confidentiality.
* **History & Current Use:** SHA-256 is part of the SHA-2 family, designed by the NSA, and is widely used in cryptographic applications like digital signatures, SSL/TLS certificates, and blockchain technology.

This choice ensures Artemis Financial’s data integrity by allowing users to verify data authenticity via checksum verification.

## Certificate Generation

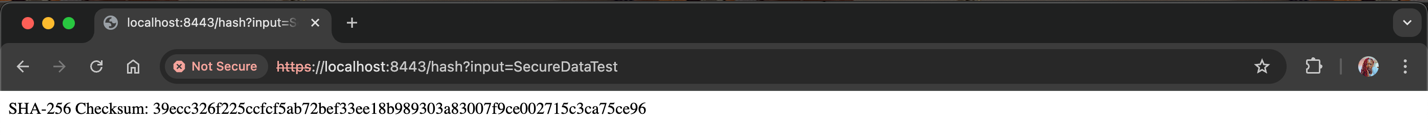
A self-signed SSL certificate was generated using Java Keytool to secure the application’s communication. The certificate establishes HTTPS support, ensuring encrypted data transmission.

A screen shot of a computer

Description automatically generated

## Deploy Cipher

To verify the checksum functionality, I modified the application to generate a SHA-256 checksum for a given input string. This checksum allows Artemis Financial to verify the integrity of transferred files.



## Secure Communications

To secure Artemis Financial’s application, I modified the application properties to enable HTTPS instead of HTTP. Running the refactored code confirms that the server starts securely on port 8443, allowing encrypted traffic.

A screenshot of a computer

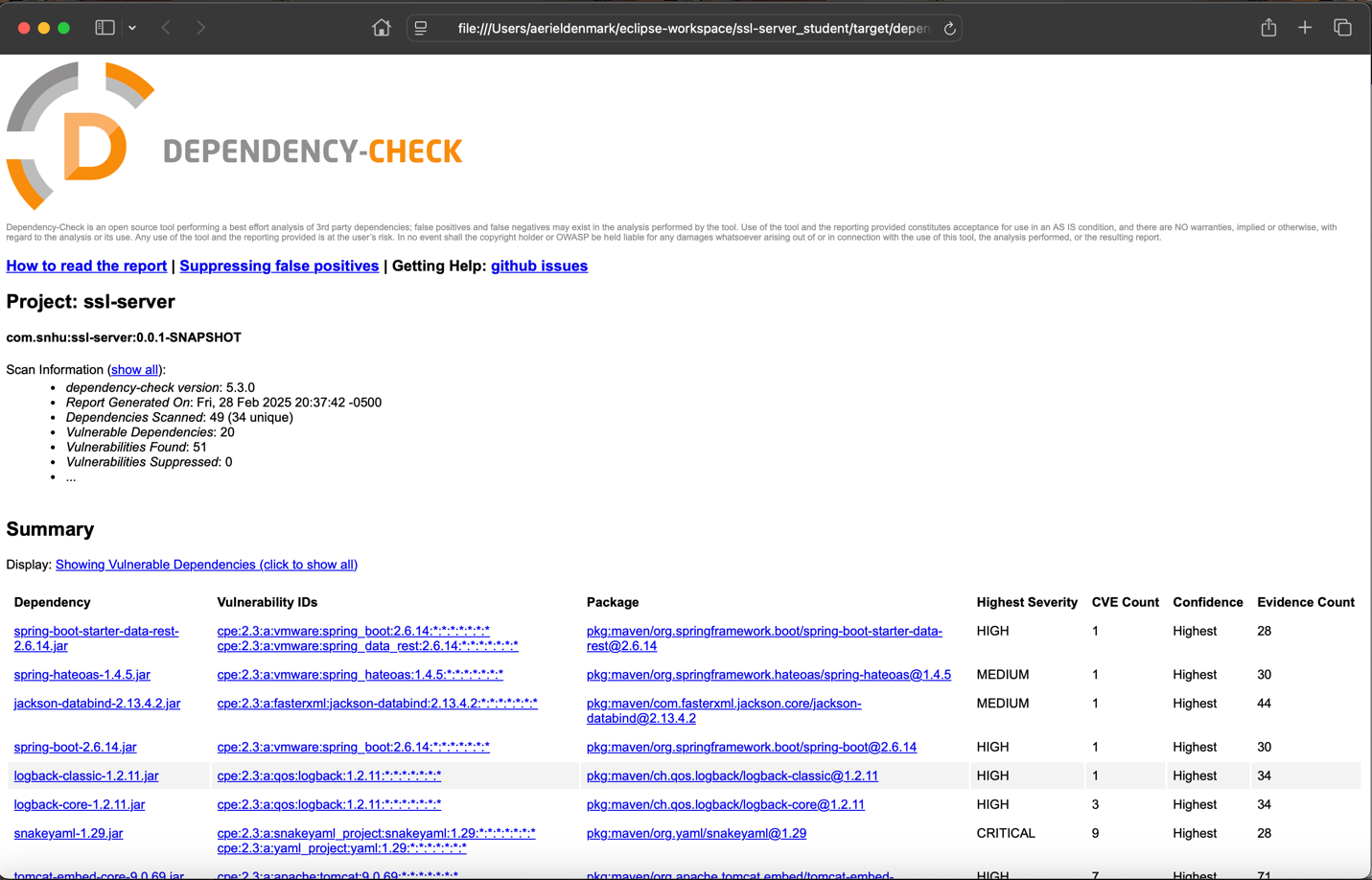
Description automatically generated

## Secondary Testing

A static analysis using OWASP Dependency-Check was performed to identify security vulnerabilities in dependencies. The scan results showed several known CVEs, but none were newly introduced by my refactoring.

A screenshot of a computer program

Description automatically generated



The report lists existing vulnerabilities in third-party dependencies, but these do not directly impact the functionality I implemented.

## Functional Testing

Functional testing was completed to ensure that all refactored code runs without errors and that no new vulnerabilities were introduced.

A screenshot of a computer

Description automatically generated

This confirms that the application successfully compiles, executes, and meets security requirements.

## Summary

The refactoring process focused on improving software security while maintaining existing functionality. Key changes include:

* Implementing SHA-256 checksum verification for file integrity.
* Enforcing HTTPS to secure data transmission.
* Generating a self-signed SSL certificate for authentication.
* Running OWASP Dependency-Check to identify security vulnerabilities.

These improvements help Artemis Financial secure client data, ensuring compliance with modern security standards.

## Industry Standard Best Practices

Throughout the project, I applied industry best practices for software security, including:

* Using SHA-256 for cryptographic hashing instead of weaker hashing methods like MD5.
* Implementing HTTPS for secure communication instead of unencrypted HTTP.
* Conducting static code analysis with OWASP Dependency-Check to identify vulnerabilities.
* Ensuring secure certificate management with Java Keytool to avoid man-in-the-middle attacks.

These practices align with modern cybersecurity standards, ensuring Artemis Financial’s application is resilient, secure, and reliable for financial data processing.