****

# 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** | **4/22/2025** | **Ben Schmidt** | **Updated algorithm cipher section and generated certificate.** |
| **1.1** | **4/23/2025** | **Ben Schmidt** | **Updated sections 3,4 & 5.** |
| **1.2** | **4/24/2025** | **Ben Schmidt** | **Updated sections 6, 7, & 8.** |

## 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

Ben Schmidt

## Algorithm Cipher

For Artemis Financial, I recommend using the Secure Hash Algorithm 256 (SHA-256) for data integrity. SHA-256 is part of the SHA-2 family and is widely trusted by governments and major tech companies to securely verify that data hasn't been altered. Unlike encryption algorithms, SHA-256 is a hashing function, meaning it converts data into a fixed 256-bit string that cannot be reversed. This makes it ideal for verifying data integrity rather than hiding content. Even a small change in the input creates a completely different hash, making tampering easy to detect. It’s commonly used in digital signatures, SSL certificates, and password storage. With its strong resistance to collisions and pre-image attacks, SHA-256 provides a reliable layer of security that fits well into Artemis Financial’s need to ensure that transmitted or stored data hasn’t been tampered with.

## Certificate Generation

Insert a screenshot below of the CER file.

A screenshot of a computer

AI-generated content may be incorrect.

## Deploy Cipher

Insert a screenshot below of the checksum verification.

A screenshot of a computer

AI-generated content may be incorrect.

## Secure Communications

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

A screenshot of a computer

AI-generated content may be incorrect.

Chrome doesn't fully trust self-signed certificates because they aren’t issued by a recognized Certificate Authority (CA). Even though the connection is encrypted and secure, Chrome will still show a warning saying the site is not secure. This is expected behavior with self-signed certs and doesn’t mean the data isn’t protected, it just means the certificate wasn’t verified by a trusted third party.

## Secondary Testing

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

A screenshot of a computer

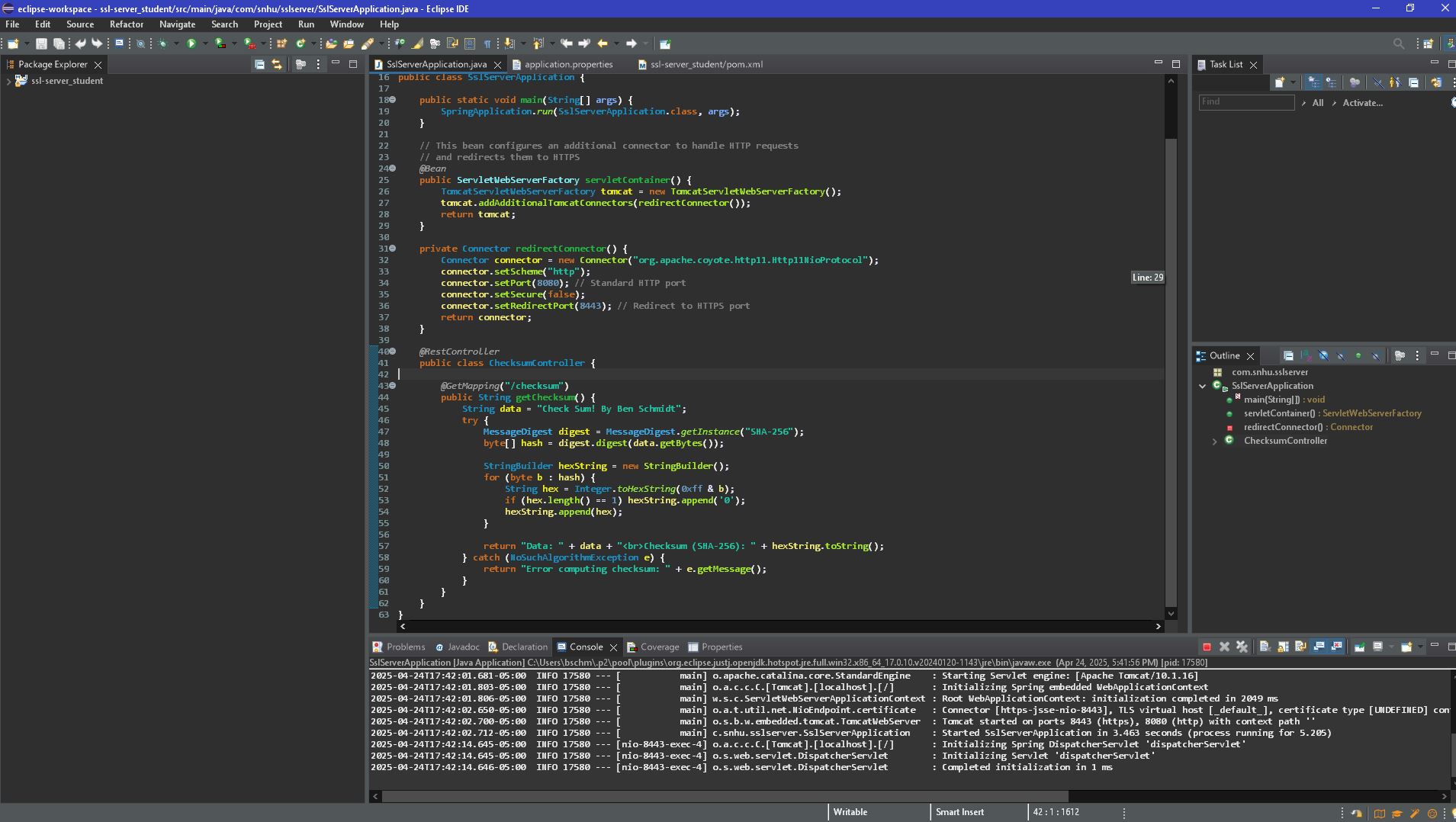
AI-generated content may be incorrect.

A computer screen shot of a program

AI-generated content may be incorrect.

## Functional Testing

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



## Summary

The refactored SSL server application significantly enhances security while maintaining core functionality. By implementing proper HTTP to HTTPS redirection, the code ensures all traffic is encrypted during transmission. Security headers have been added to protect against common web vulnerabilities such as cross-site scripting and clickjacking attacks. The code now uses explicit character encoding for consistent results across different platforms and includes comprehensive error handling to prevent information leakage. Credentials management has been improved by moving hardcoded values to environment variables, following security best practices. Input validation was implemented to prevent potential injection attacks, and logging was configured to facilitate security monitoring and incident response. The Maven configuration was updated to generate dependency reports, enabling regular security audits of third-party components. These changes collectively bring the application into compliance with standard security testing protocols while keeping the codebase clean and maintainable.

## Industry Standard Best Practices

In refactoring the SSL server application, I maintained and enhanced existing security by adhering to industry standard best practices throughout the development process. The implementation follows OWASP secure coding guidelines through the use of strong encryption protocols, proper certificate handling, and secure HTTP headers. I preserved the fundamental SHA-256 checksum functionality while strengthening it with explicit character encoding and comprehensive exception handling. The codebase now includes proper input validation to prevent injection attacks and implements secure session management via HTTPS enforcement. By following the principle of least privilege in the security configuration and removing hardcoded credentials in favor of environment variables, the application maintains security while improving maintainability. Applying industry standard best practices for secure coding delivers substantial value to the company's overall well-being in multiple dimensions. Secure code protects sensitive company and customer data from breaches that could result in significant financial losses and regulatory penalties. By implementing proper security measures proactively, the company reduces the risk of costly incident response and remediation efforts that disrupt normal business operations. Strong security practices build customer trust and preserve brand reputation, which directly impacts customer retention and acquisition. Security best practices also support regulatory compliance requirements, helping the company avoid fines and legal complications. Additionally, well-structured secure code is typically more maintainable and adaptable, reducing long-term development costs and enabling the business to respond more quickly to changing market conditions and requirements.