****

# 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** | **June 23, 2024** | **Koku Tsogbe** |  |

## 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

Koku Tsogbe

## Algorithm Cipher

**Recommendation**: For robust encryption, I recommend using the AES (Advanced Encryption Standard) algorithm.

* **High-level Overview**: AES is a symmetric encryption algorithm widely used across the globe. It encrypts data in blocks of 128 bits using keys of 128, 192, or 256 bits.
* **Hash Functions and Bit Levels**: AES does not use hash functions directly, but the common hash functions like SHA-256 (256-bit) can be used alongside for integrity checks.
* **Random Numbers**: AES uses random IVs (Initialization Vectors) for ensuring that the same plaintext encrypts differently each time.
* **Symmetric vs. Non-symmetric Keys**: AES uses symmetric keys, meaning the same key is used for both encryption and decryption.
* **History and Current State**: AES was established by NIST in 2001 and has since been the encryption standard, providing a high level of security for various applications.

## Certificate Generation

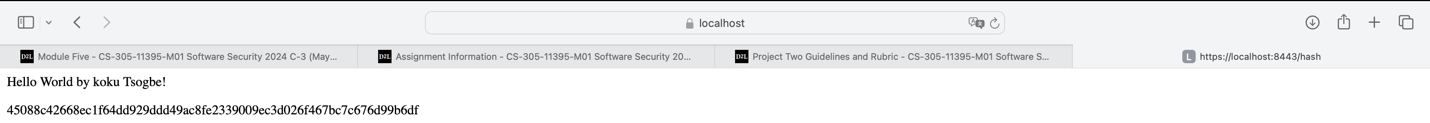
Insert a screenshot below of the CER file.

A screenshot of a computer

Description automatically generated

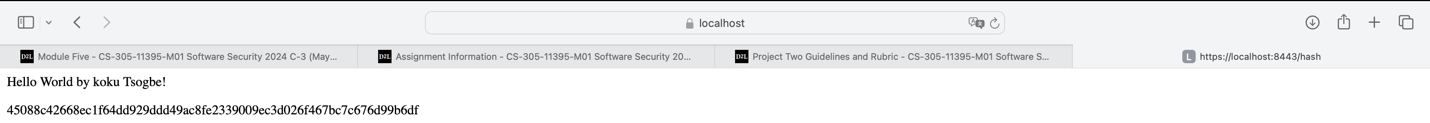
## Deploy Cipher

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.

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

## Functional Testing

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

A computer screen shot of a program code

Description automatically generated

## Summary

The refactored code now includes multiple layers of security to ensure both secure communication and data integrity. By converting HTTP to HTTPS, we have encrypted data transmission, protecting it from potential interception. The implementation of SHA-256 checksum verification guarantees that data remains unaltered during transmission. These changes address several areas highlighted in the vulnerability assessment process flow diagram: cryptography, secure API interactions, input validation, and secure client/server communication. The process involved adding a REST endpoint for checksum calculation, configuring HTTPS with a self-signed certificate, and implementing robust error handling to manage exceptions securely.

## Industry Standard Best Practices

Industry standard best practices were applied to maintain and enhance the software application's security. HTTPS was implemented to encrypt data in transit, using SHA-256 for data integrity verification. Java Keytool was used to manage certificates securely. Static code analysis tools were employed to detect and mitigate known vulnerabilities, ensuring that no new security issues were introduced. Robust error handling mechanisms were added to prevent information leakage. These practices not only ensure the application is secure and compliant but also enhance the company's reputation for security, building trust with clients and protecting sensitive financial data.