****

# 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** | **10/17/24** | **Daniel Vidmar** | **Initial Work** |

## 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

Daniel Vidmar

## Algorithm Cipher

**Overview of AES**: Advanced Encryption Standard is a symmetric encryption algorithm widely used to secure data. It encrypts data blocks using a single key applied in encryption and decryption processes. AES supports three key sizes: 128, 192, and 256 bits, with more extended key sizes providing more robust encryption. Due to its efficiency and security, AES is the standard for data encryption across many industries.

**Hash Functions and Bit Levels**: Although AES is an encryption method, it is often used with cryptographic hash functions like SHA-256 to verify data integrity. SHA-256 generates a fixed 256-bit hash from any input data, ensuring that even a tiny change in the input will result in a drastically different hash output. This makes it ideal for detecting data tampering or corruption during transmission or storage.

Random Numbers, Symmetric vs. Asymmetric Keys:

* **Random Numbers**: AES requires random values known as initialization vectors (IVs) to ensure that the resulting ciphertext will differ even if the same data is encrypted multiple times. The quality of the random number generation plays a crucial role in the strength of the encryption.
* **Symmetric vs. Asymmetric Keys**: AES is a symmetric encryption algorithm that uses the same key for encryption and decryption. This is different from asymmetric encryption algorithms like RSA, which use a pair of keys for encryption and decryption. Symmetric algorithms like AES are generally faster and more efficient than asymmetric algorithms, making them suitable for encrypting large amounts of data.

**History and Current Use**: NIST introduced AES in 2001 to replace the older DES (Data Encryption Standard), which had become vulnerable to attacks. Thanks to its balance of performance and security, AES has become the dominant encryption standard used by governments, financial institutions, and businesses worldwide. It remains highly effective against modern attack methods and is widely implemented in hardware and software solutions.

## Certificate Generation

Insert a screenshot below of the CER file.

A screenshot of a computer program

Description automatically generated

## Deploy Cipher

Insert a screenshot below of the checksum verification.

A screenshot of a computer code

Description automatically generated

## Secure Communications

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

A screenshot of a computer

Description automatically generated

## Secondary Testing

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

A computer screen shot of a program code

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

In the refactored code, several steps were taken to comply with security testing protocols and ensure the application remains secure. As outlined in the provided flow diagram, the vulnerability assessment process guided the focus on key security areas, including data encryption, secure communication, and checksum verification. By refactoring the code to implement secure HTTPS communication and adding checksum verification using cryptographic hashing (SHA-256), the application better protects sensitive data and resists tampering or unauthorized alterations during transmission.

The process of enhancing security in the application involves multiple steps. First, I implemented HTTPS using SSL certificates to encrypt communications between the server and the client. This layer of protection ensures that data transmitted over the network is encrypted, making it difficult for attackers to intercept or manipulate. Second, I added cryptographic hash functions to generate checksums, which act as a verification layer to maintain data integrity.

## Industry Standard Best Practices

I adhered to industry standard best practices for secure coding when refactoring the code. This involved using robust encryption algorithms such as AES and SHA-256, widely recognized as secure and effective. I also implemented secure communication protocols like HTTPS to ensure encrypted data in transit. Additionally, using Java Keytool to generate self-signed certificates ensured that the application complies with SSL standards for secure communication.

Applying these best practices safeguards the application against known vulnerabilities and maintains the integrity and confidentiality of user data. Following these standards minimizes the risk of potential attacks, such as man-in-the-middle or replay attacks. Ensuring secure coding practices contributes to the company's overall well-being by fostering trust with users, protecting sensitive information, and avoiding costly security breaches that could damage the company's reputation and finances.

**Resources**

Robin, E. (2023, December 1). *Best and free tools for 256-bit AES military-grade encryption*. Newsoftwares.net Blog. https://www.newsoftwares.net/blog/best-and-free-tools-for-256-bit-aes-military-grade-encryption/

Zavak, D. (2024, April 19). *What is AES encryption and how does it work?*. WEZOM. https://wezom.com/blog/what-is-aes-encryption-and-how-does-it-work