****

# 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** | **8/17/2024** | **Linkhon Hasan** | **Final Report** |

## 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

Linkhon Hasan

## Algorithm Cipher

1. I recommend the “256” variant of the SHA-2 algorithm, which takes data in the form of bytes and converted them to a 256 bit hash. This makes the 256-bit hash impossible to be converted back to its original state and this is very desirable as it ensures data security. It also helps prevent collision, which means producing the same hash value for two different data sources and defeating the purpose of hashing and maintaining data integrity in the process.
2. The purpose of hash functions is to ensure data integrity by taking in normal data and turning it into a string of random bytes, this protects the data from being manipulated or even being read if accessed illegally; while higher bit levels can be seen as longer string of those randomized bytes, it makes it harder to attackers to decrypt data.
3. The use of symmetric keys needs less compute power, which is useful for encrypting a large chunk of data and only a single key is used to encrypt and decrypt the data. Whereas asymmetric key algorithms have a public key and a secret key, and data transaction only occurs when the key pair matches. Random numbers and their generation is very important as they ensure that the encryption sequences created via our security algorithms are truly random and unpredictable; thus making them impossible for hackers to crack.
4. One of the earlies ciphers developed and widely used was the Caesar cipher and the DES. DES was invented around the 1970s and made use of symmetric keys. They are both considered deprecated and not secure due to them being very easy to crack by hackers. In modern times, hash methods/functions are widely used as they offer very good security and the use of quantum computers in the future to implement and strengthen cryptographic ciphers will bring forth a new generation of algorithms and new standards, along with their own risks and challenges.

## Certificate Generation

Insert a screenshot below of the CER file.

A computer screen with white text

Description automatically generated

A screenshot of a computer

Description automatically generated

## Deploy Cipher

Insert a screenshot below of the checksum verification.

A screen shot of a computer

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 screenshot of a computer program

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 screenshot of a computer program

Description automatically generated

A screenshot of a computer

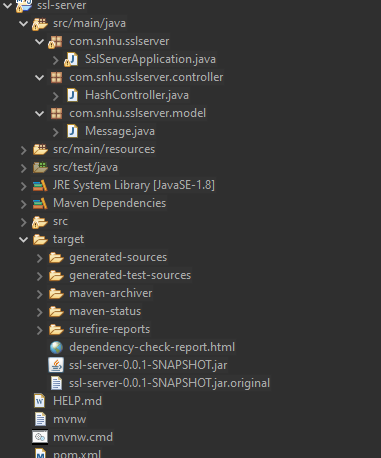
Description automatically generated

A screen shot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated



## Summary

In order to make the application comply with security standards as shown by the Vulnerability Assessment Flow, I have addressed a lot of potential security issues. They are as follows:

* I have moved the API into a separate controller, this will help in implementing input and data validation within the application.
* I have implemented a hashing algorithm that encrypts data being served by the application, thus enhancing security.
* I have moved the message into a separate model class that makes use of encapsulation, this will prevent unintended and malicious modification of data.
* I have used an external dependency tool to ensure the external libraries used by our cipher algorithm do not open our application to attacks due to new insecure dependencies.
* I have also enabled HTTPS protocol for our application so that the API handles secure interactions via web.

## Industry Standard Best Practices

I made use of encapsulation of data, which is an industry standard to keep data hidden away from being accessed in an unintended manner. I have also broken down the application to follow a three-layer structure where different parts of the app are in separate packages, which follows the abstraction methodology of object-oriented programming. I have also conducted secondary and functional testing which helps ensure the app functions properly and does not have any severely vulnerable dependencies. Lastly, I have enabled the use of secure protocol from communication since it is a web application. Following industry best practices for secure coding is best for the company’s well being as it protects the company from attacks that may lead to loss of sensitive data, which may lead to lawsuits. It also helps the company be in line with government regulations, which helps avoid any unnecessary issues with regulatory agencies.