****

# 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** | **12/15/2024** | **Justin Schumann** |  |

## 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

Justin Schumann

## Algorithm Cipher

[Insert text.]

Since our client’s success depends on their customers’ trust, we would need a cipher with a proven track record. SHA-256 is the ideal choice, since it is widely used and has a long-documented track record. SHA-256 is nearly impossible to reverse-engineer, because it is a one-way function. Another pro is Collison resistance. Collisions could give hackers the opportunity of providing them an data set that happens to produce the same value as another data set, SHA-256 prevents that from happening.

The best way to describe Hash Functions is to imagine them as a list of directions given to a collection of data, in order to change into a fixed-length string that is referred to as a “hash value”. Hash functions are mathematical operations. In Computer Science, a bit is the smallest unit of information. The bit level of SHA-256 is 256 bits, or 32-bytes (a byte is 8 bits). That means that it produces a 256-bit hash value. SHA-256 uses a symmetric-key, a single key instead of a pair of keys (Asymmetric). Though asymmetric key encryption is more secure, using a symmetric one is faster and more efficient. For all keys, both symmetric and asymmetric, the number used to generate a key, is always random. Using a random number instead of a determined one keeps it more secure and less guessable.

The history of encryption is long, being recorded throughout human history. It was in the 1970s, with the 1977 release Data Encryption Standard (DES) by both IBM and the U.S government that encryption started to play a major role in computer science. Though encryption like devices were used greatly in WW2, and developed sometime before, it was DES that set the standard. Now computers and data having an increasing role in our daily lives, with all information even so sensitive being digitalized, encryption is more necessary than ever.

## Certificate Generation

Insert a screenshot below of the CER file.

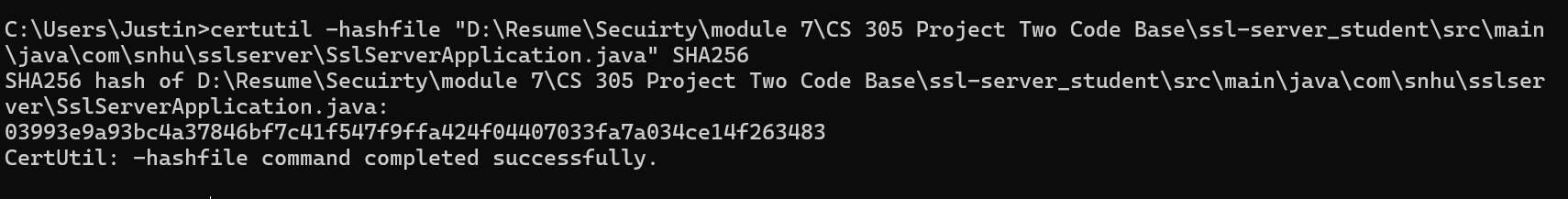
[Insert screenshots here.]



## Deploy Cipher

Insert a screenshot below of the checksum verification.

[Insert screenshots here.]



## Secure Communications

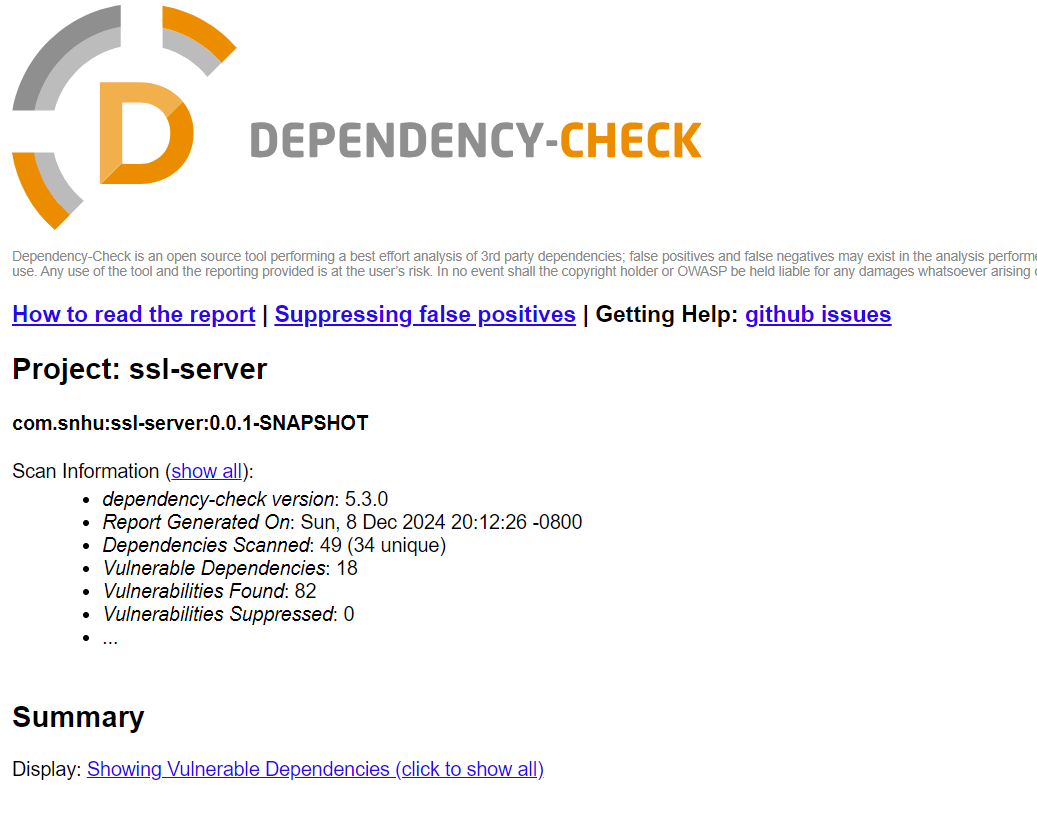
Insert a screenshot below of the web browser that shows a secure webpage.

[Insert screenshots here.]

## Secondary Testing

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

[Insert screenshots here.]



## Functional Testing

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

[Insert screenshots here.]

## Summary

[Insert text.]

As for the reasons already mentioned, I have chosen HA-256 as my hash. I did some edits to the SSLSApllication.java, adding a RequestMapping and RestController. The dependency check on the pom.xml was updated to the latest version from 5.3.0 to 11.1.1. In sum, I made all the necessary changes that I could.

## Industry Standard Best Practices

Like any professor, I try to do my best to maintain I.S.B Practices. For communications lines to be secured, HTTPS was used. OWASP dependency check was employed, to analyze vulnerabilities. Data that was sensitive was encrypted. To have a collision resistant, fast computing cipher, SHA256 was chosen. Along with using a hashing algorithm, password policies were used to have safe user access.

Employing Industry Standard Best Practices secures that our project is both safe and trusted enough to be used. Ignoring them could mean not gaining the trust of our customers. Even if it is more easy in the short term, preserving our company’s public image would be more profitable in the long term.

Sources:

Types of Encryption Algorithms + Pros and Cons for Each. KeyFactor.com.

<https://www.keyfactor.com/education-center/types-of-encryption-algorithms/>

Hash Functions and Types of Hash functions. (2024, May 29). GeeksForGeeks.com.

<https://www.geeksforgeeks.org/hash-functions-and-list-types-of-hash-functions/>