****

# Practices for Secure Software Report

**Rebekah Collett**

**2/19/2023**

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** | **2/19/2023** | **Rebekah Collett** | **Refactored Code as well as implemented more secure protocol** |

## Client



## 

## Developer

Rebekah Collett

## Algorithm Cipher

For an encryption algorithm cipher that avoids collisions we would recommend using the SHA-256. (“Class MessageDigest”, 2020)

Collisions are important to avoid since it can make two inputs have the same encryption. This causes weaknesses in the security and can make it so hackers can worm their way into the system. (Lake, para. 10)

Collisions invalidate the security of the certificate causing it so other may be able to access your information. This is also backed by the fact that there are 2256 possible hash values when using it, the likely hood of documents having the same exact hash value is extremely rare, if possible, at all. (Callaghan, para. 4).

The 256 of the cipher refers to how many bits are used in the cyphered text. This makes it so more calculations must be done to enable the cypher to be broken.

There are components needed to be able to generate the cypher. Since the cypher requires keys, a random number generator is needed to generate the key. (Smirnoff & Scholten, 2019)

Encryptions are nothing new. There is a rich history of them being used, especially during times of war. This can be traced back clear to the Spartans. (Staff, para. 2) these were done on paper and only recently have we, as a society, been able to implement encryptions in a digital form. The first standard was the Data Encryption Standard, or DES. It formed in 1973. This remained uncracked until 1997 where it was replaced with the Advanced Encryption Standard, or AES. This is still used as the standard for high encryption level. ((Staff, para. 7)

## Certificate Generation

CER File

Text

Description automatically generated

## Deploy Cipher

Checksum Verification

Text

Description automatically generated with medium confidence

## Secure Communications

Secure Website

Graphical user interface, text, application, website

Description automatically generated

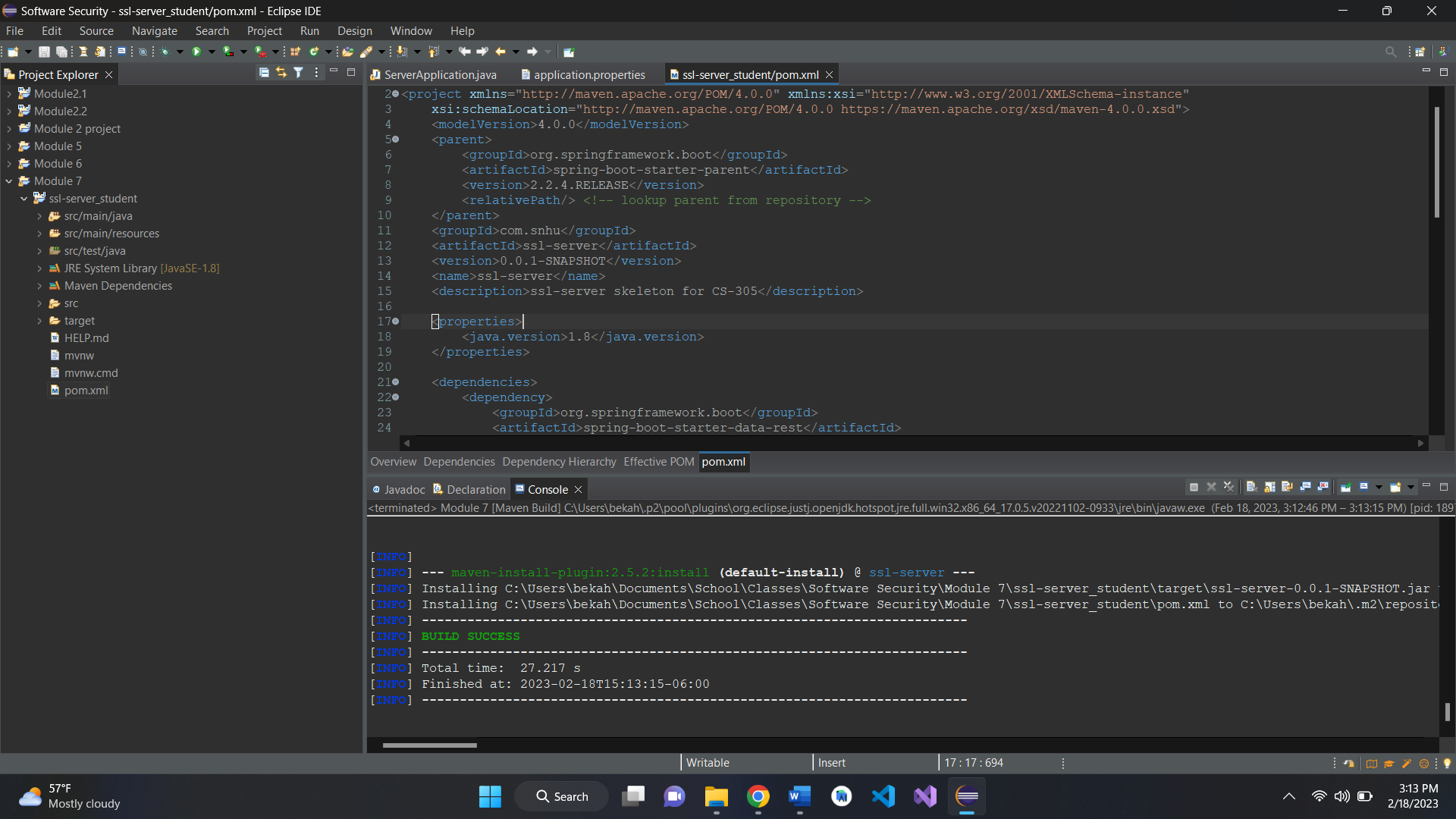
## Secondary Testing

Refactored Code Executed Without Error

A screenshot of a computer

Description automatically generated

Dependency-check Report



## Functional Testing

Refactored Code Executed Without Errors

A screenshot of a computer

Description automatically generated

## Summary

Since I did not get any new errors when changing the Http to Https, I did not need to do any refactoring. The code stayed the same and was able to work with the certification and algorithm.

When first writing the code I made sure to address the areas on the Vulnerability Assessment Process Flow Diagram to enable it to be secure. The areas of vulnerability that I addressed were API, Input, and client/server.

The input was address by adding the cypher encryption in. This included generating a certification key as well as inputting a bit function that would be able to manipulate the string that was inputted. The securing of adding Https vs Http was added as well. The helped make the website itself more secure.

## Industry Standard Best Practices

The biggest practice that I used when refactoring this code was to secure the input. I made sure there was an exception handler to catch improper input. This helps unwanted visitors from accessing areas of the website they are not supposed to.

I also ran a dependency check to make sure that there were not any know vulnerabilities in the dependencies that the software is using.

By addressing even just these few practices, it helped the program be a lot less vulnerable to attacks. This means that the clients can have more faith in the company. This also means the company can work on bettering their software instead of having to go back and fix past code.

Citation

Callaghan, P. (n.d.). Why you should use SHA-256 in evidence authentication. Pagefreezer Blog.

Retrieved February 5, 2023, from <https://blog.pagefreezer.com/sha-256-benefits-evidence->authentication#:~:text=Collisions%20are%20incredibly%20unlikely%3A%20There,the%20exact%20same%20hash%20value.

Lake, J. (2022, March 30). *What is a collision attack?* Comparitech. Retrieved February 5, 2023, from https://www.comparitech.com/blog/information-security/what-is-a-collision-attack/

Smirnoff , P., & Scholten , U. (2019). *The role of random number generators in relation to HSMS & Key Management*. Cryptomathic. Retrieved February 18, 2023, from https://www.cryptomathic.com/news-events/blog/the-role-of-random-number-generators-in-relation-to-hsms-key-management#:~:text=The%20role%20of%20RNGs%20in,as%20initialization%20vectors%20and%20nonces.

Staff. (2020, June 24). Class MessageDigest. Messagedigest (Java Platform SE 7 ).

Retrieved February 5, 2023, from

Thales Group. (2023, February 1). *A brief history of encryption (and cryptography)*. Thales Group. Retrieved February 18, 2023, from https://www.thalesgroup.com/en/markets/digital-identity-and-security/magazine/brief-history-encryption