****

# 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** | **12/8/23** | **Cooper Brien** |  |

## Client



## Developer

Cooper Brien

## Algorithm Cipher

The Algorithm Cipher that I will be deploying is SHA-256. SHA-256 is a hash function with hash values that are the size of 256 bits. The function is collision resistant, meaning that it is difficult for hackers to access the data within. For these reasons, avoiding collisions is important for keeping data integrity and security. SHA-256 is also used widely by many cryptocurrencies for verification purposes as a result of its high collision resistance. This cipher’s hash functions help keep the information that is being encrypted secure. Using the hash functions to encrypt the data makes it nearly impossible to brute force. The has functions also pad the data that it is encrypting to bit levels of 256 bits. SHA uses symmetric key encryption, meaning that there is only one key that is used to both encrypt and decrypt data. Encryption algorithms need to be constantly monitored and updated in the case that they get exploited and broken through. There are many encryption algorithms that once worked before but have now been long exploited by hackers. It is important to make sure you are using up-to-date encryption methods with no known exploits to better protect your data.

1. **Certificate Generation**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

[Insert screenshots here.]

## Deploy Cipher

A screenshot of a computer

Description automatically generated

## Secure Communications

I tried many different methods to get the browser to trust the certificate that I generated. I tried adding it to my trusted certificate authorities, but it still says the local site is unsecure. Below is a screenshot to prove that the application is at least using my self-signed certificate, but it still says unsecure.

A screenshot of a computer

Description automatically generated

## Secondary Testing

A screenshot of a computer

Description automatically generated

Below is a screenshot of the suppression.xml file used to suppress the false positive vulnerabilities.

A screenshot of a computer

Description automatically generated

## Functional Testing

Below is a screenshot of the code that I added to the application to verify the checksum. There are no errors within the code and no errors on startup. There was an error that I fixed within the project by updating the Maven Dependencies. Once everything was properly updated and installed, the errors were resolved.

A screenshot of a computer

Description automatically generated

## Summary

During this project, I spent most of my time resolving issues surrounding cryptography, code errors, code quality and secure communications between client and server. There were some syntactical errors in the code base that were mainly caused by out-of-date libraries, or they were trying to import from libraries that were not installed. To fix these issues I had to make sure that my libraries were up to date an installed correctly. To solve the issues with cryptography, I had to implement an algorithm cipher that would be able to encrypt and decrypt messages sent through the application. Finally, I had to generate a self-signed certificate to ensure that the communication between the client and server was secured. The process for adding layers of security was very systematic and required a lot of testing. I usually focused on implementing one security measure at a time and then testing to make sure it was functioning properly.

## Industry Standard Best Practices

I used industry standard practices by employing defensive programming skills, along with error handling while also employing constant tests throughout development to ensure that the program was functioning correctly. There is great value in using industry standard practices when developing any project. Constantly testing your code can help reduce errors and the time it takes to debug. Using conventional naming standards accompanied by frequent comments makes the code easier for others to read and understand.