****

# 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 5](#_Toc102040760)

[4. Secure Communications 5](#_Toc102040761)

[5. Secondary Testing 5](#_Toc102040762)

[6. Functional Testing 6](#_Toc102040763)

[7. Summary 7](#_Toc102040764)

[8. Industry Standard Best Practices 7](#_Toc102040765)

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.1** | **12/15/1024** | **Alexander Ray** |  |

## 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

[Insert your name here.]

## Algorithm Cipher

[I have chosen to use the SHA-256 algorithm for this project. Like all other secure hash algorithms, SHA-256 employs irreversible hash functions, so when the desired data gets encrypted within the program there is no way for a third party to ever be able to reverse engineer the encrypted data without access to the necessary authorization keys to do so. The NSA themselves had a direct hand in the creation of the SHA-256 algorithm and continue to use it to this day, so the validity of its protection capabilities cannot be denied. I have selected SHA-256

Hash functions and bit levels aid the program in determining key sizes, returns, object placements and so on in order to ensure that the desired dimensions and functionalities are included in the encryption process. It is through these precise and meticulous means that the algorithm’s key dimensions are decided. Then the chosen variables get organized where the keys can be delegated, recorded and returned etc. which is how the encryption within the system can be situational.

The point of a cipher is to encrypt the data located within by using the hash method to assign randomized characters to represent the data that is being encrypted, so that anyone from the outside looking in will only see a jumbled mess of numbers and letters. The keys SHA-256 used are symmetrical, so both the encryption and decryption functionalities take the same key input. This is as opposed to a non-symmetric cipher that creates a different key for each functionality. There are advantages and disadvantages to both methods, with the symmetrical’s use of only a single key for its functionality making it easier to keep up with and use, but also ensures that if the key ever gets out then the entire project is compromised. On the other hand, the multiple keys for the non-symmetrical cipher means that there is more information to keep up with, but also that losing one key doesn’t automatically mean that the entirety of the project is compromised. There is also the fact to consider that once any key is lost or misplaced, whether it by symmetrical or non-symmetrical, that functionality will be locked and unreachable. This is why much consideration should be used when

## Certificate Generation

Insert a screenshot below of the CER file.

A computer screen with white text

Description automatically generated

[Insert screenshots here.]

## Deploy Cipher

Insert a screenshot below of the checksum verification.

A screenshot of a phone

Description automatically generated

## Secure Communications

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

[Insert screenshots here.]

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.

[Insert screenshots here ]

A screen shot of a computer

Description automatically generated

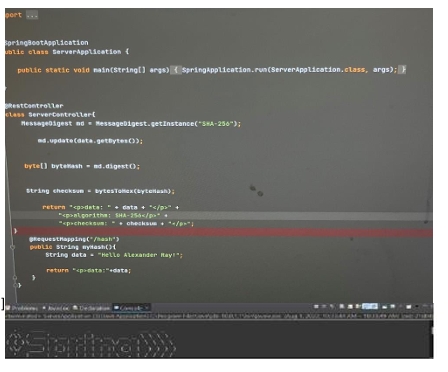
<file:///C:/Users/alexa/AppData/Local/Temp/%7B3F2250E6-7BC3-4110-8596-780960B3735A%7D/%7B75385FC5-E40D-4B6B-963E-1C44BACA7EA9%7D/dependency-check-report.html>

A screenshot of a computer error

Description automatically generated

## Functional Testing

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

[Insert screenshots here.] 

## Summary

We were tasked with refactoring and testing a code that is meant to use algorithmic hash functions to encrypt, contain, and organize the data. First, we had to select a cipher algorithm, and I went with SHA-256 due to its delicate balance between complexity and ease of use. Through the use of this algorithm the code will then encrypt and assign a key to the desired data, with that key being used to both encrypt and decrypt the data. We then ran dependency checks on the Pom.xml files to ensure that there were no false positive vulnerabilities, and by creating a separate suppression.xml file we had to suppress all the false vulnerabilities and refractor the Pom.xml file to include these changes. By doing this we ensure that the code is not wasting energy and effort on issues that don’t even exist in order to maximize efficiency. We also had to use the command prompt to create a password and input our data that would be used for the cipher. Finally, we had to run the refactored code and create a secure HTML link that would output our data and provide an encrypted message containing the “sensitive” data.

## Industry Standard Best Practices

Encrypted algorithms, not surprising, grew along with the rise of the civilian internet in the early 2000s as the growing demand for online protection increased. As we can clearly see the internet in the current day is bigger and wider reaching than ever before, which means that encryption algorithms are right there with it. With the increase of technology and overall societal computer literacy, there has naturally been an increase in the capabilities of hackers and malicious entities and programs that wish to take advantage of the ever-growing online community. For this reason, it is important that the technology and innovators at the forefront of the industry, as well as the countess individuals who work directly within the industry to ensure not only their own protection, but also the protection of their clients and users.