****

# Practices for Secure Software Report

**Table of Contents**

[**Document Revision History 3**](#_30j0zll)

[**Client 3**](#_1fob9te)

[**Instructions 3**](#_3znysh7)

[**Developer 4**](#_2et92p0)

[**1. Algorithm Cipher 4**](#_tyjcwt)

[**2. Certificate Generation 4**](#_3dy6vkm)

[**3. Deploy Cipher 4**](#_1t3h5sf)

[**4. Secure Communications 4**](#_4d34og8)

[**5. Secondary Testing 4**](#_2s8eyo1)

[**6. Functional Testing 4**](#_17dp8vu)

[**7. Summary 4**](#_3rdcrjn)

[**8. Industry Standard Best Practices 4**](#_26in1rg)

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **Oct 7, 2023** | **Matthew Courts** |  |

## 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

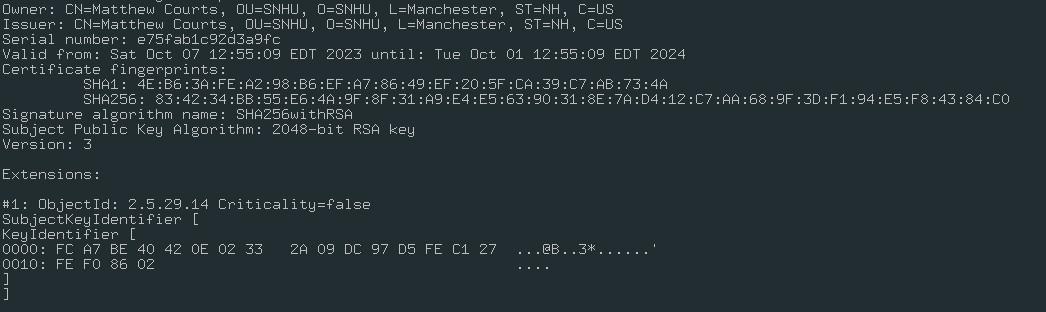
Matthew Courts

## Algorithm Cipher

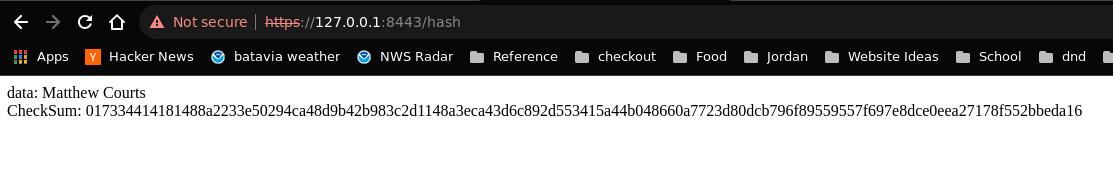
The Cipher Chosen is the SHA-512 a member of the SHA-2 family that is a part of the currently recommended hashing algorithms by the National Institute of Standards and Technology (NIST). Superseding SHA-1 which NIST is removing out of the recommended list by the year 2030. Other common ciphers are not currently up to date or recommended by the NIST, MD5 used to be a majorly used cipher that was recently removed. Blowfish was a largely used cipher that is now out of date, the creator recommends Twofish for x64 based computers as blowfish was developed for 32bit with a 64 bit block limitation. AES replaced DES due to being stronger and easier to implement. RC5 has been replaced by RC6 allowing for integration with the AES standardization process. AES is commonly used but integrates well with other forms of ciphers allowing for stacking of cipher algorithms in higher security needs for communication and or storage. RCA and DSA are used for communication encryption using a public and private key system to validate received data.

Cipher algorithms have various moving parts that allow them to produce the encrypted and unencrypted data. Hash functions create a standard size of block data for the encryption to be consistent. This occurs by breaking the source data down to block sizes, for AES this is 128 bits. No block will ever be more or less than 128 bits allowing for consistency with the bit shift and matrix algorithm to be applied without error. Keys are created and depending on the use case of the cipher this could be symmetrical or non-symmetrical keys. Symmetrical are much longer than their counterpart and are used for encryption and decryption this can be used for detecting alteration along a message transportation but can leave the key to the public eye. The counterpart to the non-symmetrical is a pair of keys one public and the other private. The private key is kept hidden for decryption while the public is exposed outside of the system. Both keys are mathematically related and can be checked against each other before decryption.

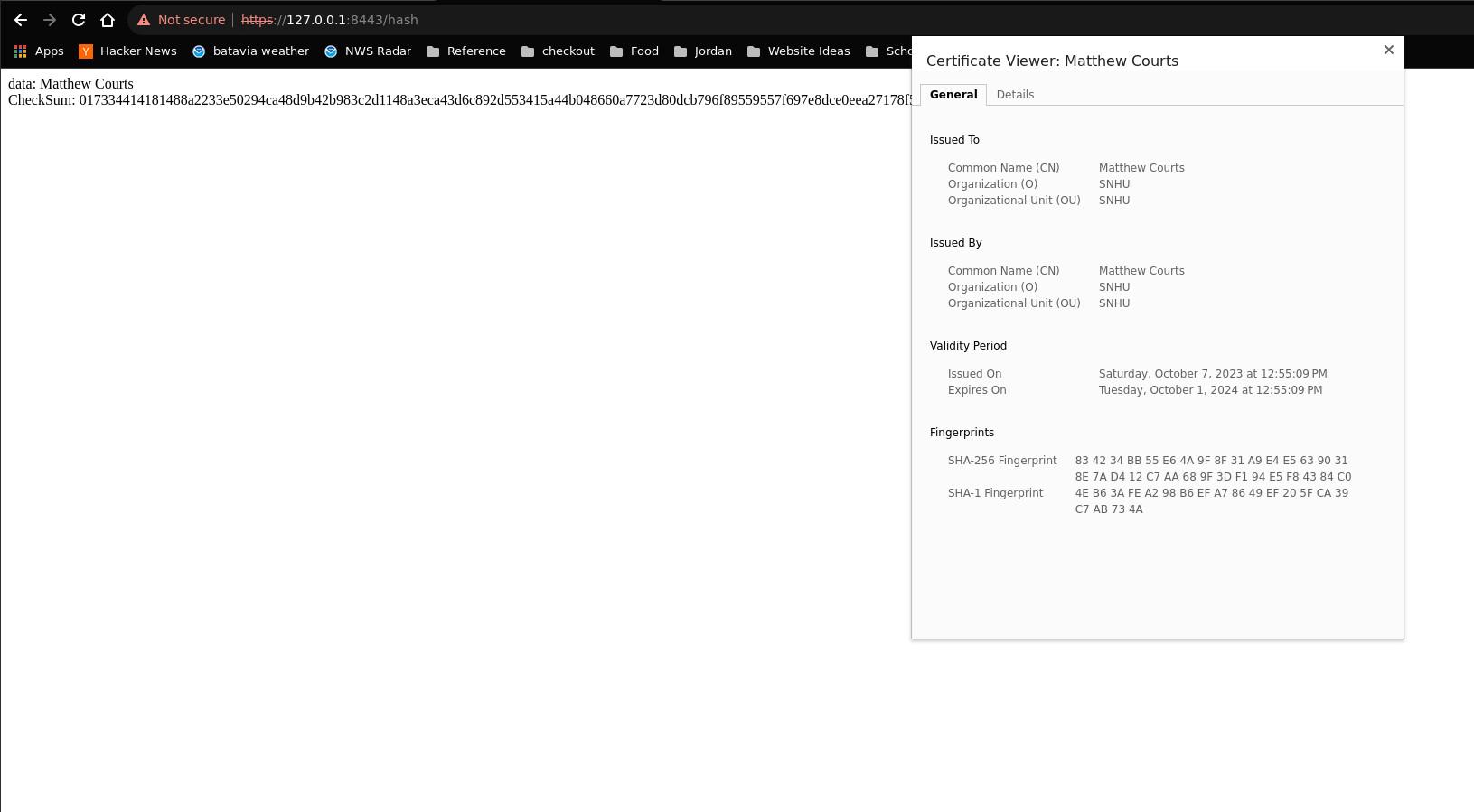
## Certificate Generation



## Deploy Cipher



## Secure Communications



## Secondary Testing

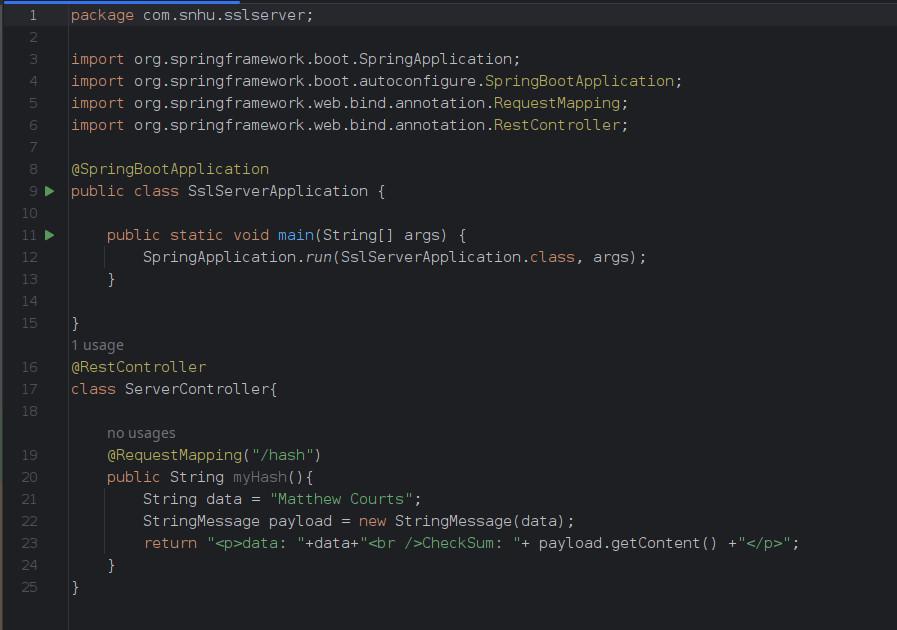
Before code refactored (any framework based vulnerabilities are ignored and outside the scope of the assignment):

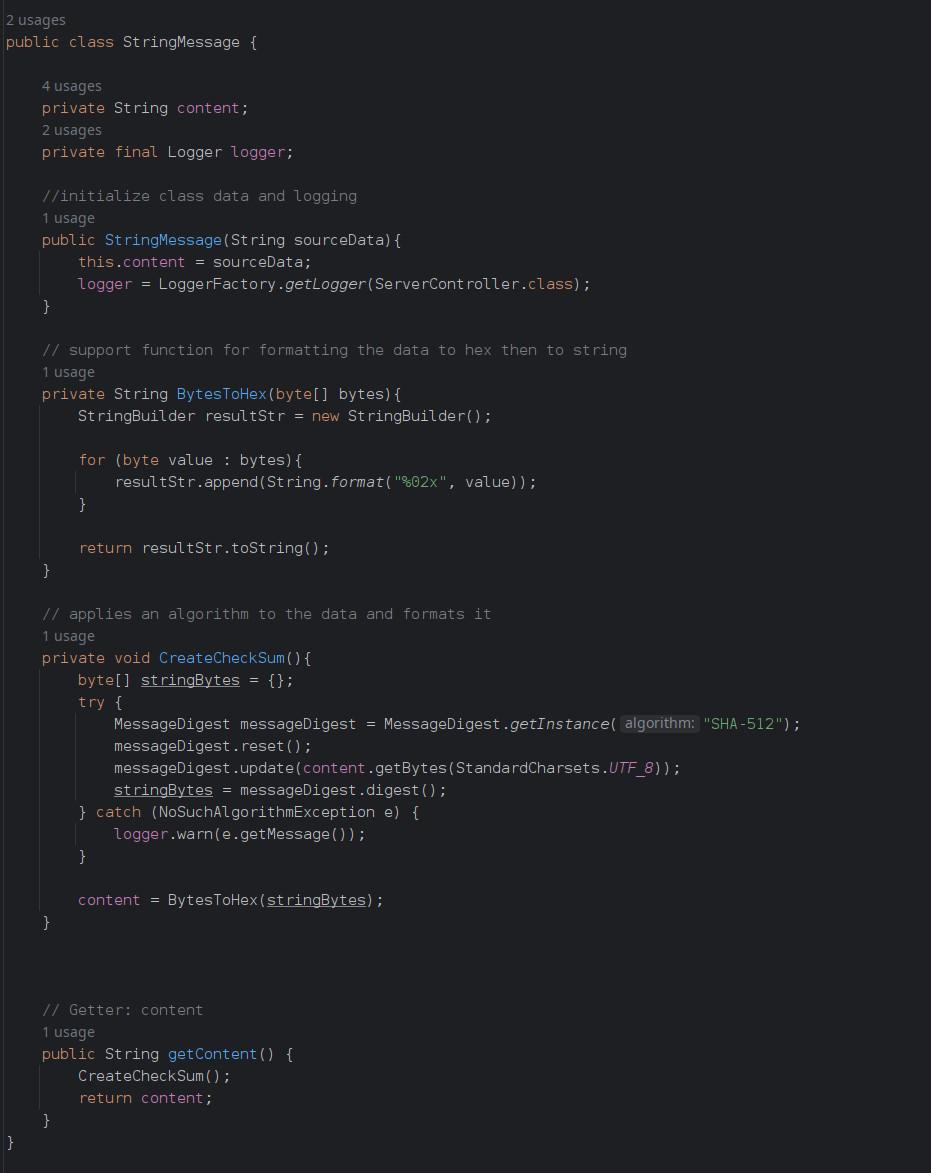


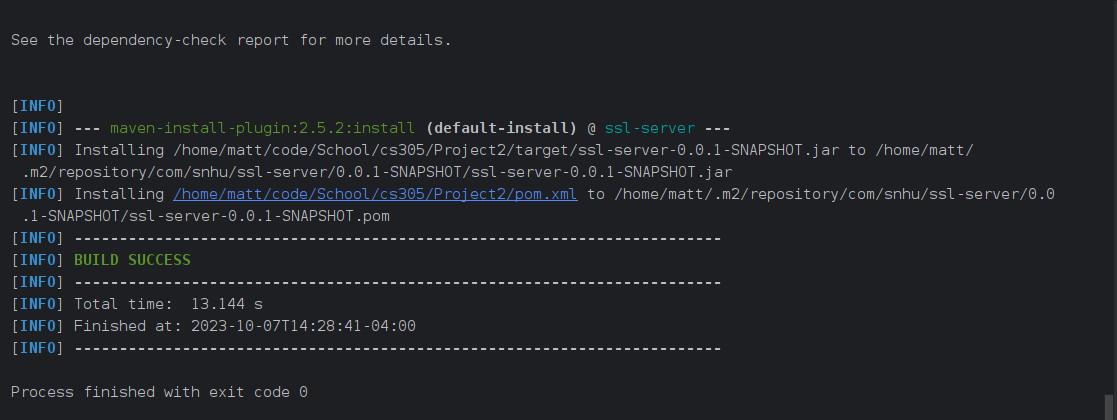
After refactoring:



## Functional Testing







## Summary

Vulnerability assessment areas require review to determine what parts of the software need to be reviewed and or secured. With Global Rain the area of focus are Secure APIs, Cryptography, Client/Server, and Code Error. With the current code structure being built on a Model, View, Control (MVC) framework this utilizes means of client server communication via an API control structure. The means for a secure and safe connection to the server needs to utilize encryption and error handling to handle different encountered use cases of the service. Upon adding a response route in the controller this created a point of access for the client to the API structure. Communication should be encrypted leading to SSL security implemented and configured with a certificate. Further more data was encapsulated into a response class that encrypted the response content with SHA-512 preventing a middle man entity from viewing the data. Logging is also added to maintain any errors during the encryption allowing active monitoring and response when an issue arises. All this required vulnerability assessment application to the Model, Controllers, Views, and API areas of the framework.

## Industry Standard Best Practices

Adding new features the software for Global Rain required securing communication, data, and ensuring the new feature does not create a new security hazard. Utilizing the vulnerability assessment and dependency state checker the new features can be validated to create a safe and secure operating environment for the clients connecting to the service. With the dependency state checker all software packages that create the spring framework used are checked for any known CVEs that are known vulnerabilities for the software used. This allows for correcting and known vulnerability before deployment. Adding the features used there was no change in known Vulnerabilities within the scope of the requested features.

Applying industry recommended standards allows for creating an up to date safe operating environment and sets the baseline for how sensitive data should be handled. There are too many vectors for attacks on data and services. Using industry tools and reports allows for companies to address new features and value propositions being explored or released without damage to the users or consumers. These tools allow for companies to create a cyclical development once released to maintain, monitor and improve in a reactive manner when a threat does occur.