****

# Practices for Secure Software Report

**Table of Contents**

[**Document Revision History 3**](#_heading=h.30j0zll)

[**Client 3**](#_heading=h.1fob9te)

[**Instructions 3**](#_heading=h.3znysh7)

[**Developer 4**](#_heading=h.2et92p0)

[**1. Algorithm Cipher 4**](#_heading=h.tyjcwt)

[**2. Certificate Generation 4**](#_heading=h.3dy6vkm)

[**3. Deploy Cipher 4**](#_heading=h.1t3h5sf)

[**4. Secure Communications 4**](#_heading=h.4d34og8)

[**5. Secondary Testing 4**](#_heading=h.2s8eyo1)

[**6. Functional Testing 4**](#_heading=h.17dp8vu)

[**7. Summary 4**](#_heading=h.3rdcrjn)

[**8. Industry Standard Best Practices 4**](#_heading=h.26in1rg)

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **6/18/2023** | **Amanda de los Santos** |  |

## 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

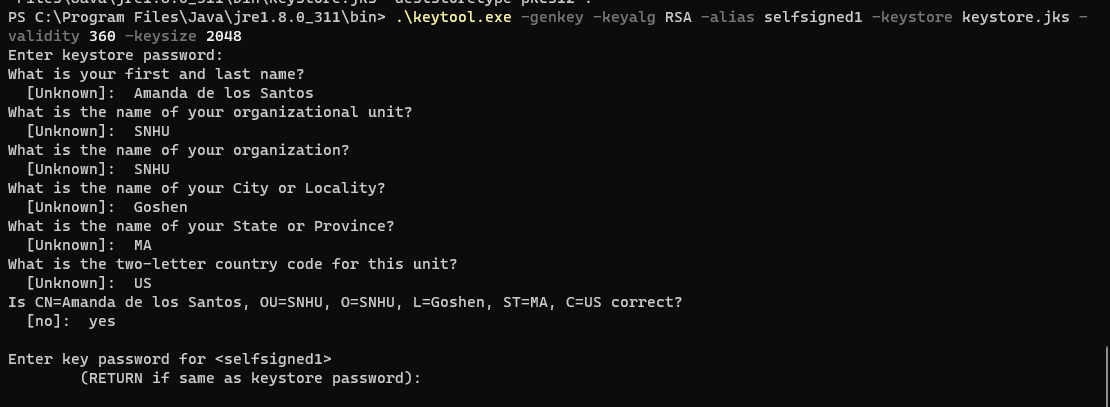
Amanda de los Santos

## Algorithm Cipher

The recommended algorithm cipher for this application is SHA-256. SHA, or Secure Hash Algorishm, was created by the National Security Agency. SHA-256 specifically is a cryptographic hash function that outputs values that are 256-bits long. SHA-256 is one of the most secure hash functions out there right now. This is one that the US government requires agencies to use to protect the sensitive information they have. This cipher can withstand most brute-force attacks as it would need 2^256 attempts to generate the initial data.

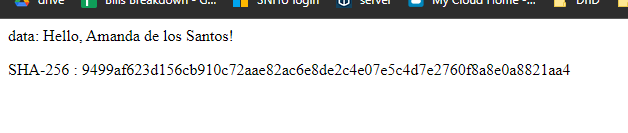
## Certificate Generation

Insert a screenshot below of the CER file.



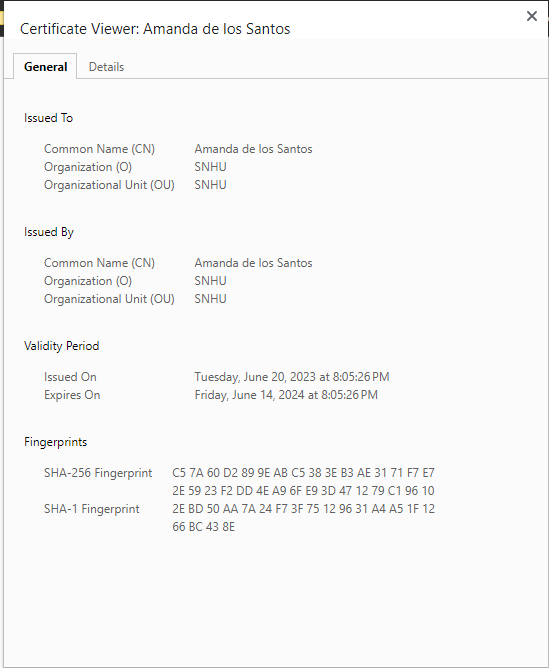
## Deploy Cipher

Insert a screenshot below of the checksum verification.



## Secure Communications

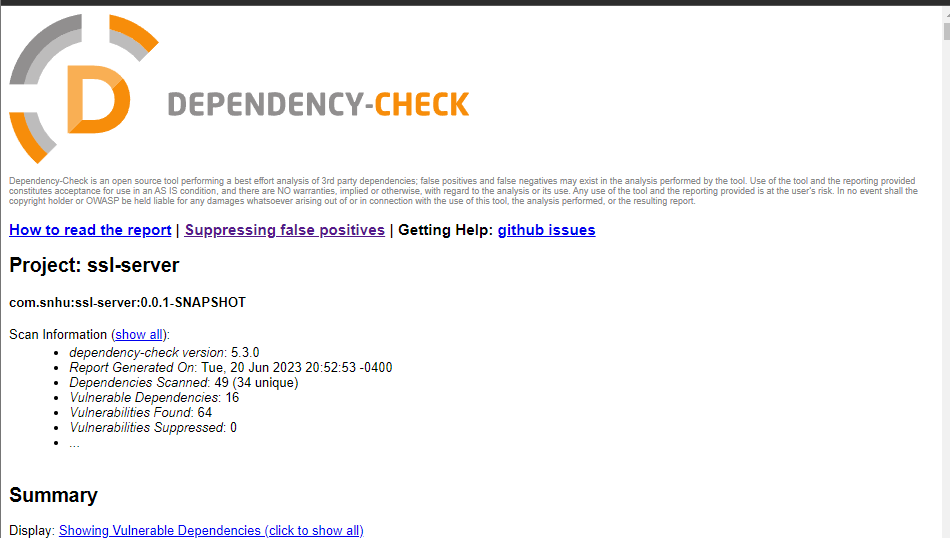
Insert a screenshot below of the web browser that shows a secure webpage.

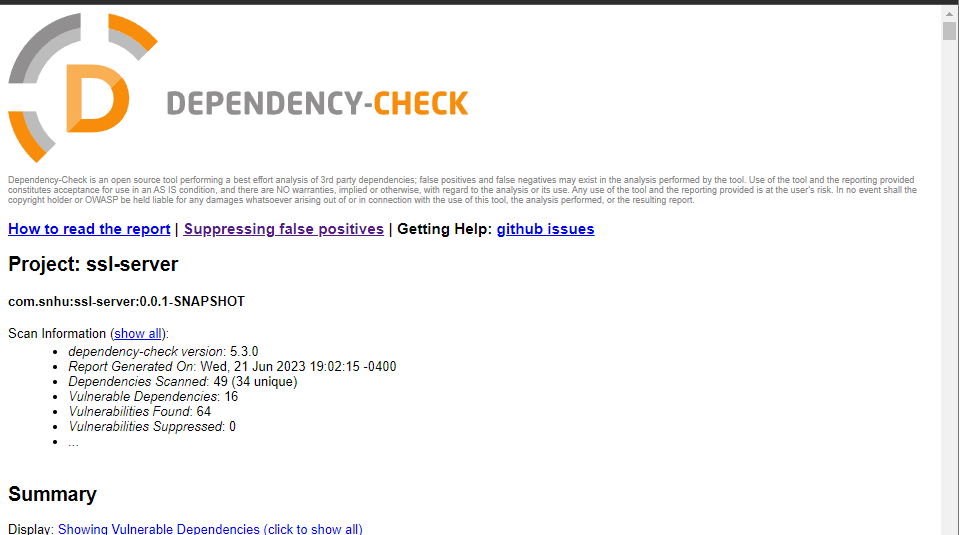


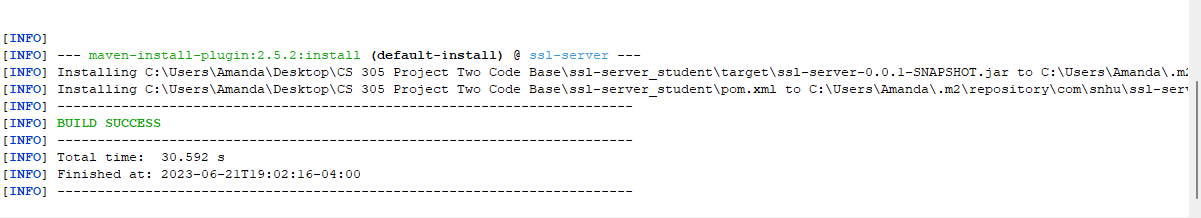
## Secondary Testing

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

Initial dependency check:

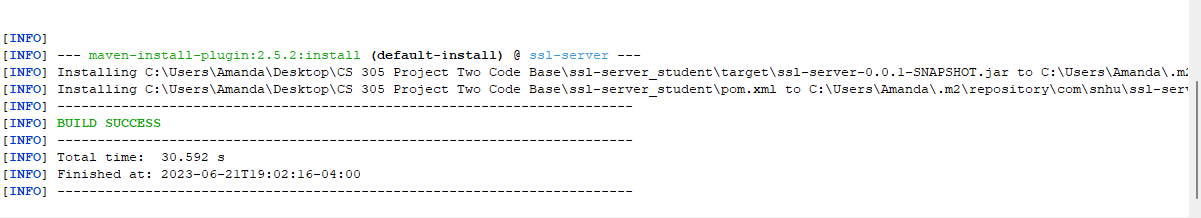
  
  
  
After refactoring:





## Functional Testing

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



## Summary

Refactoring code is a positive step in security as it helps strengthen any vulnerabilities that may be in the application. This will create more peace of mind for the company knowing that the sensitive data within the application is better protected. Using the algorithm cipher described above, SHA-256, would protect the application against many types of attacks, such as brute-force attacks. Continually updating and checking the application for new updates for any of the libraries within the application will also help protect the application, as well as continuing to check for new threats that have arisen.

## Industry Standard Best Practices

Keeping in mind the current industry standard is key to continual improvement in terms of securing an application such as this one. Making sure that the most recent and best algorithm cipher will make sure all data remains secure constantly. The developers working on the application should make sure the code remains clean and secure throughout building the application, and while checking for any vulnerabilities. They should also be continually testing for new or existing security issues.