****

# 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** | **12/07/2023** | **Joseph Kawamoto** |  |

## 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

Joseph Kawamoto

## Algorithm Cipher

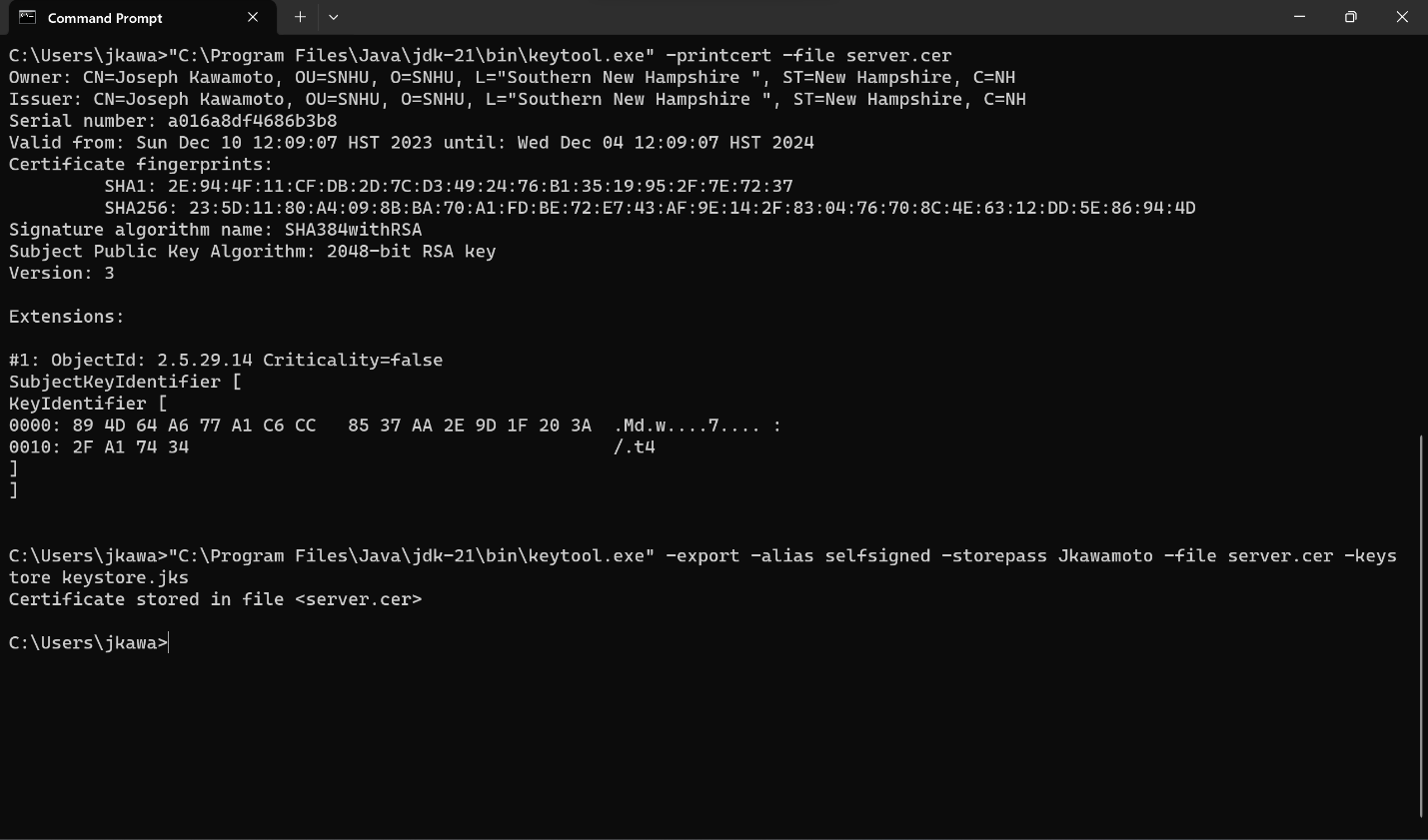
I again opted to use the SHA-256 hash algorithm as my algorithm cipher. The main reason that I selected this hash algorithm is because of the very low likelihood of collision as well as the robust security features. This is because of the large output space that SHA-256 uses. The range of potential hash values is 2^256 which makes it nearly impossible that two separate strings of data will end up with the exact same hash values. SHA-256 also stands as a benchmark for the industry and is recognized for its high-end security attributes.

Random numbers play are used to create unique keys and vectors. The unpredictability of random numbers enhances the security of cryptographic operations which can prevent attackers from predicting the values of the key or picking up on patterns.

Symmetric key algorithms use the same key for both encryption and decryption. Sharing the key between two parties poses a security problem, but symmetric keys are very efficient. Asymmetric key algorithms use a pair of keys, public and private. The public keys are shared but the private keys are kept secret. When data is encrypted with the public key, it can only be decrypted by using the matching private key in the pairing. This makes exchanging keys between two parties much simpler than symmetric keys.

Encryption and ciphers have been used for hundreds of years, for example the Ceasar cipher which would move letters back a pre-determined amount. Since then, data encryption and decryption have become much more sophisticated and are used more widely now than ever. Today, encryption is a very important part of digital security, keeping sensitive data such as finance, communication, and personal data safe from malicious actors. As technology has advanced, so too has encryption methods.

## Certificate Generation

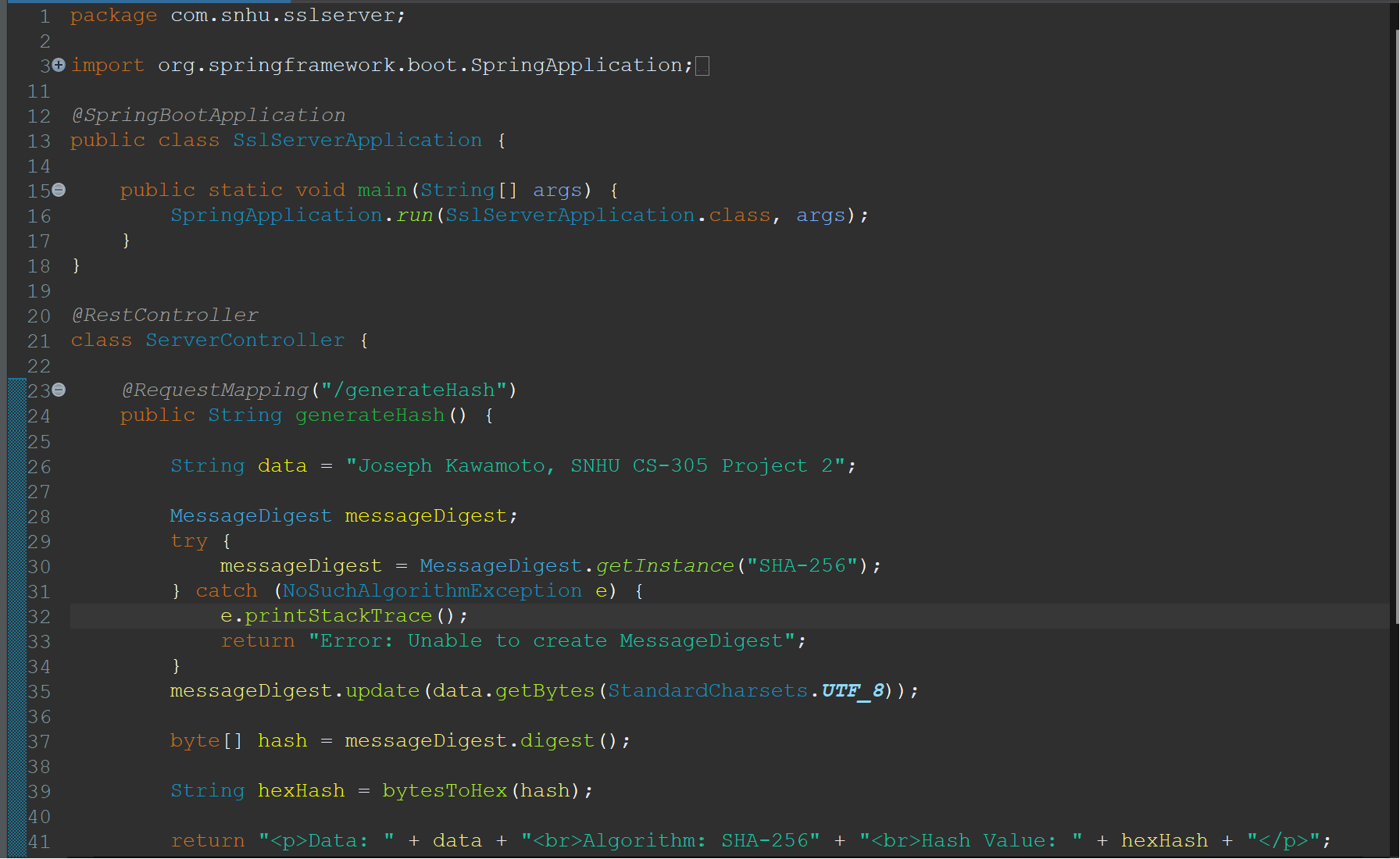
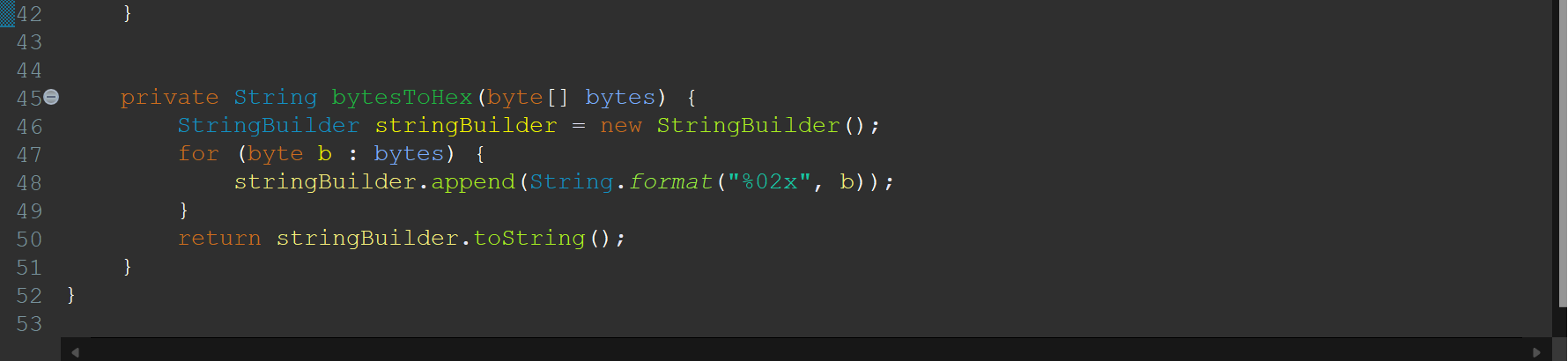


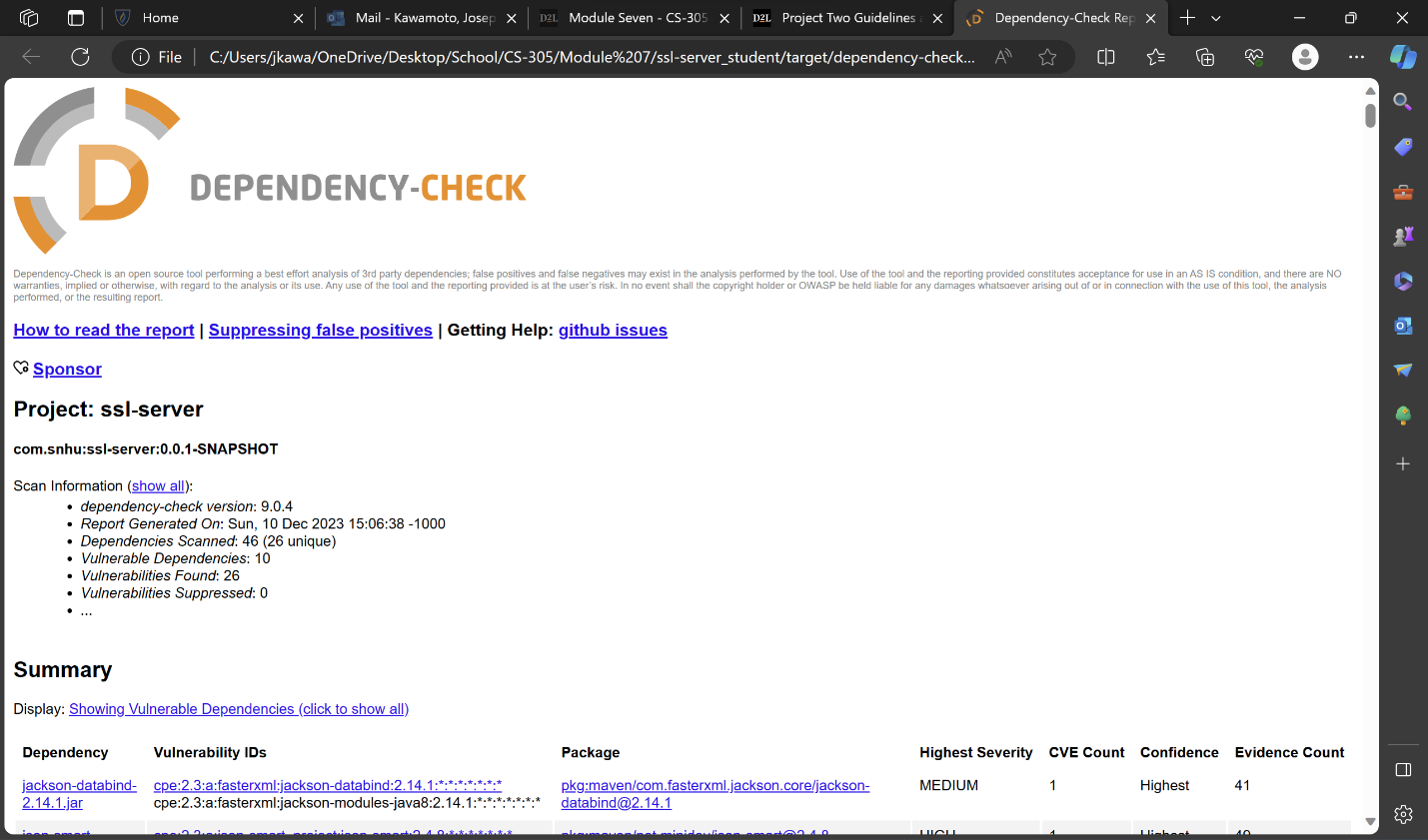
## Deploy Cipher

## Secure Communications

Unfortunately, I could not figure out how to secure my project. I tried installing the certificate into the “Trusted Root Certification Authorities” by clicking on the server.cer file that I created, and I also tried to add the certificate to my browser directly but nothing worked. I’m not sure what I am doing wrong, but I was unable to complete this step.

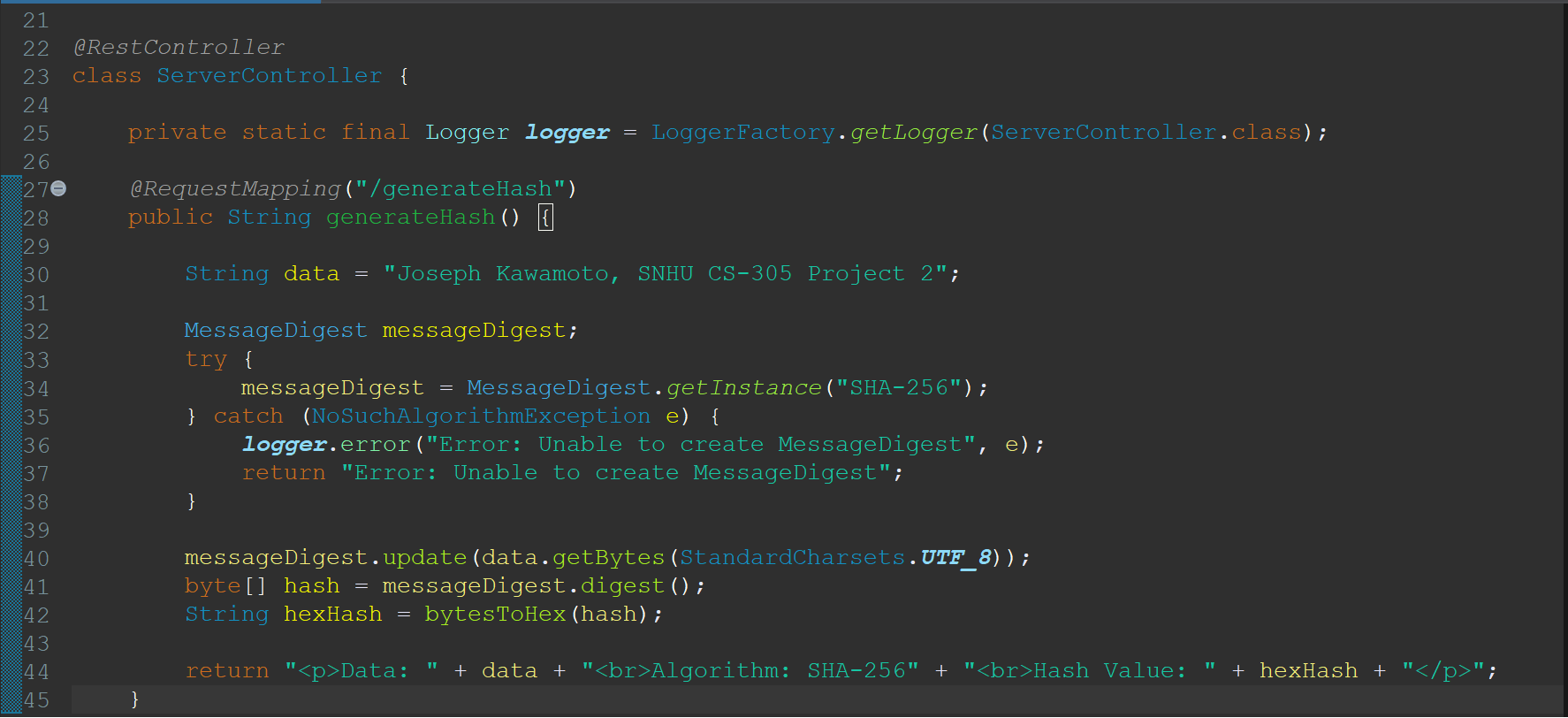
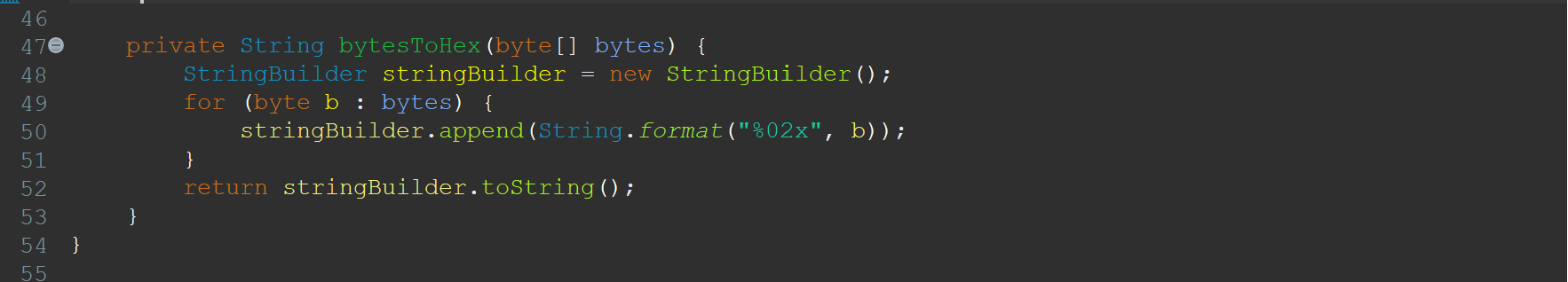
## Secondary Testing

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



## Functional Testing





## Summary

My refactoring focused on the hashing process and exception handling. To ensure that the hashing was secure, the code uses the SHA-256 algorithm, which is a very popular and widely used hash function. This choice is highlighted in the part of the code where the “MessageDigest” instance is initialized with “SHA-256.” Also, the “NoSuchAlgorithmException” prints an error message telling the user something is wrong rather than printing the stack trace. This is important for preventing security risks because it avoids exposing sensitive information through leaking. In the POM file, I updated the OWASP dependency check to the latest version. I also updated dependencies that had newer versions where necessary.

## Industry Standard Best Practices

The use of the SHA-256 hashing algorithm is in line with industry standard best practices for choosing a secure algorithm. This hashing algorithm is widely recognized for its strength and resilience against attacks. Also, the code has proper exception handling. Instead of printing stack trace directly to the console, the code uses SLF4J for logging, which aligns with industry best practices.

Implementing industry standard best practices is important for safeguarding the well-being of the company. By implementing standardized and recognized security protocols and best practices, the company can significantly reduce the risk of data breaches and attacks. Also, having a proactive approach to handling security measure instills trust in the customers and stakeholders, which can enhance the reputation of the company and boost their credibility. Most importantly, using secure coding practices is important in meeting regulations which can help keep the company out of legal trouble. By constantly updating and keeping on top of industry standard security measures, the company can more effectively respond to new threats and minimize disruption.