****

# 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** | **[Date]** | **[Your Name]** |  |

## 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

Chandni Singh

## Algorithm Cipher

SHA 256 would be the most useful and efficient encryption algorithm for Artemis Financials’ archived files. This is due to several reasons. The first is that it provides frequent software updates and training – as having our employees up to date on new features can save the company from potential attacks. My next reasoning is because cracking SHA 256 is extremely difficult due to its large number. Because of this, SHA 256 would make a very safe algorithm against security attacks of any sort. In all honestly, I don’t see any risks or downfalls when using SHA 256. When researching SHA 256 and governmental regulations I’ve observed that the NIST has some sort of issues with SHA 256 and recommends government agencies to utilize other algorithms. Although they recommend stopping the use of SHA 256 – at the same time they say that SHA 256’s hash functions has a very secure algorithm.

The implementation of she 256 is that it has patent encryption, and it generates a 256 bit value. When data is encrypted, the reader cannot decrypt the encryption without a key. This key makes SHA 256 of the most secure cyphers that a lot of companies and agencies are utilizing to this day – which would be perfect for Artemis Financial.

The most secure cypher out there in my opinion is the AES cypher but it has a few drawbacks such as: it is hard to implement, its very complex and SHA 256 can still be used more quickly and efficiently to retrieve archived flies due to how advanced AES algorithms is.

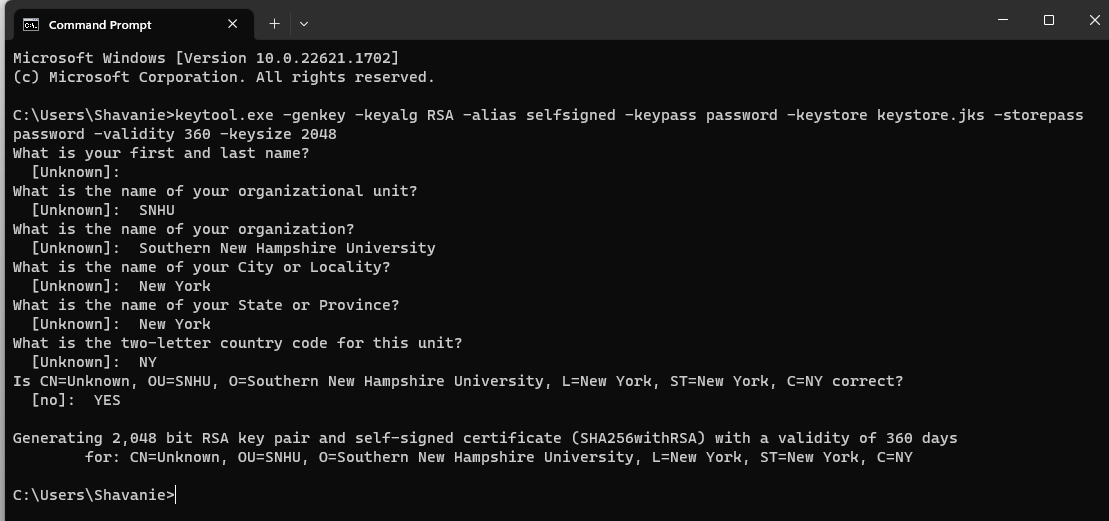
The purpose of hash functions is that it is useful to define data or any size and that it can be found in many applications that deal with security. The function is used to map data of any size to any fixed-size values. This is called hashing to scatter storage addresses. The hash algorithm uses a map of binary strings to encrypt and decryption data – for this more cryptographic hash functions are created to take strings of any length as input as it produces a fixed length hash value.

The use of random numbers requires the users to cover the binary number with zeros. The user continues to do this action of converting the input until the size is equal to 512. For the use of asymmetric and symmetric encryption both use different techniques. In asymmetric encryption we use a single key that is given out to all the receivers of the message ( this encryption in particular allows for two separate keys – a kay to decode and a kay to encode the data. For symmetric encryption, electronic data is encrypted and decrypted using one single key. This key must be altered between the encoder and decoder to encrypt and decrypt the message. Although this sounds like a lot more work, symmetric encryption is faster than the latter.

The history and current state of encryption algorithms starts with cyphers. The cypher that first has been recorded to be made was by Vigenere in the 16th century and he even made an encryption key. Cyphers is an algorithm that is used to encrypt and decrypt data. To protect the key, the encryption algorithm and the cypher text are used as covers – which has been a practice for centuries. For our current state of encryption, we must keep in mind that technology is developing faster than ever, which means that security risks are also developing at the same rate. At the same time there are various advanced algorithms like SHA 256 to address these security threats and secure data and archived files.

## Certificate Generation

Insert a screenshot below of the CER file.



## Deploy Cipher

Insert a screenshot below of the checksum verification.

A picture containing screenshot, text, software, display

Description automatically generated

## Secure Communications

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

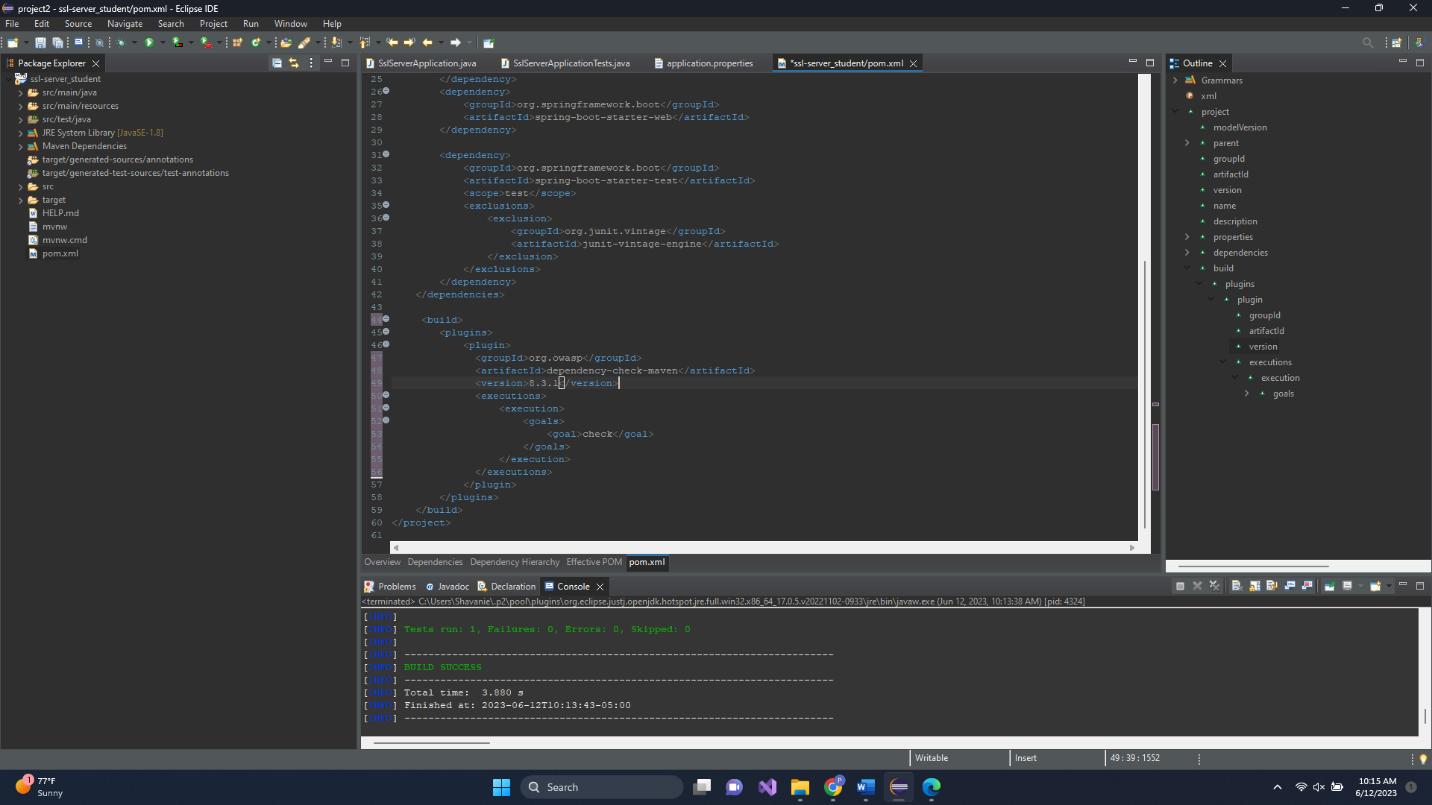
A picture containing text, software, screenshot

Description automatically generated

## Secondary Testing

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

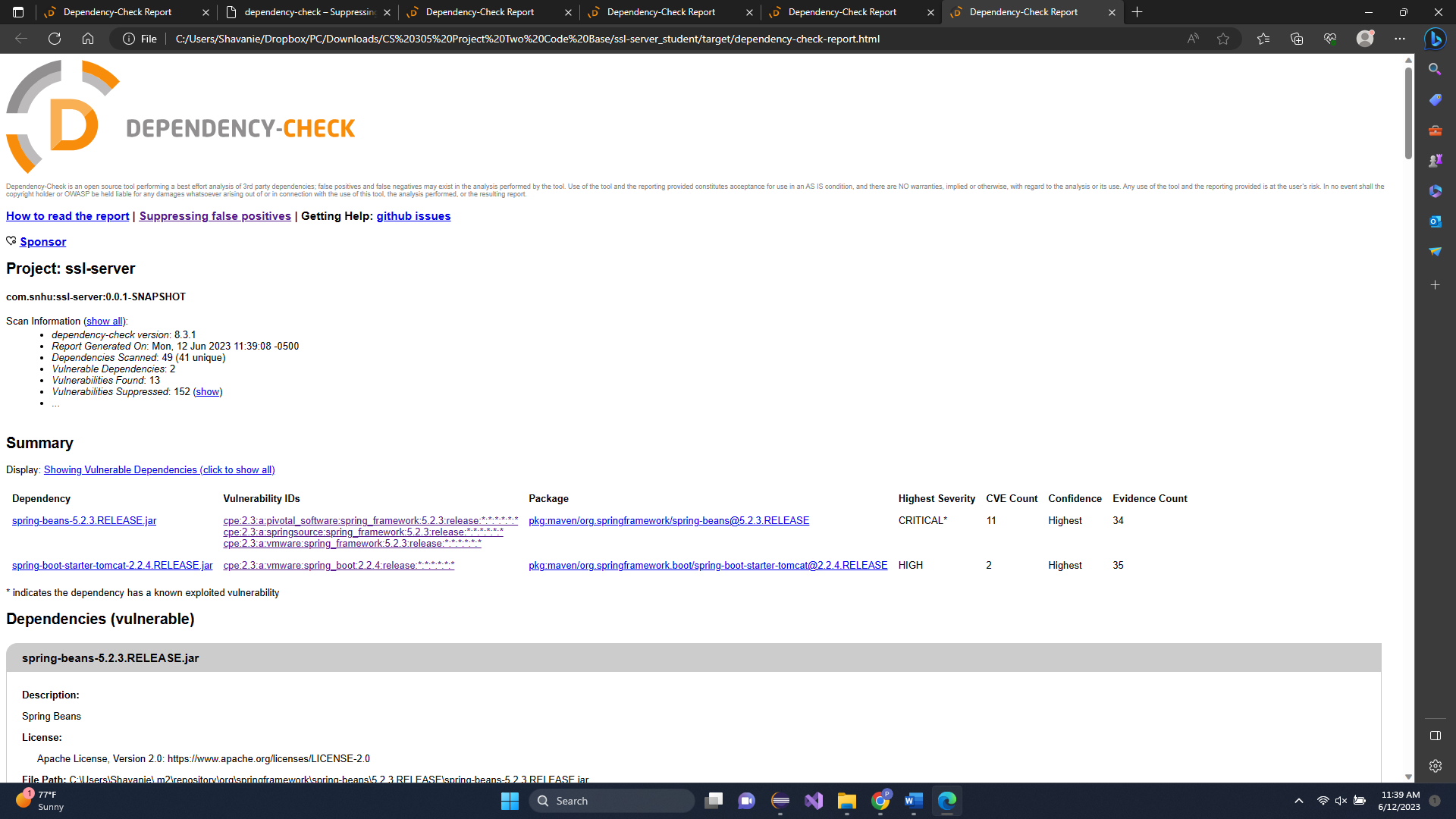
Before suppressions:



A screenshot of a computer

Description automatically generated

After Suppressions:



## Functional Testing

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

A computer screen shot of a program

Description automatically generated with low confidence

## Summary

My code was refactored by first updating the pom.xml file to the latest dependency check version. Then I added a suppression.xml file and copied and pasted the suppressions from the first run. I did a total of four 3 runs, the first coming up with 100 suppressions’, the second, 16, the third 14 and the fourth run with 13. Following the Vulnerability assessment process flow diagram, I revised the initial code, I checked for errors, then I continued to check the new dependency check results from each cycle.

## Industry Standard Best Practices

I applied industry best practices for secure coding to mitigate known security vulnerabilities by updating the current code in the pom.xml, to the latest dependency check code. Using updated code helps to keep the company’s overall wellbeing secure because it forms a wall against any new security risks and outside data violations. Keeping the company, its employs, and users/customers safe from hackers or data breaches is important for the company to prosper and for everyone to be safe and secure.