****

# 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** | **8/12/2024** | **Buxton McCaslin** |  |

## 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

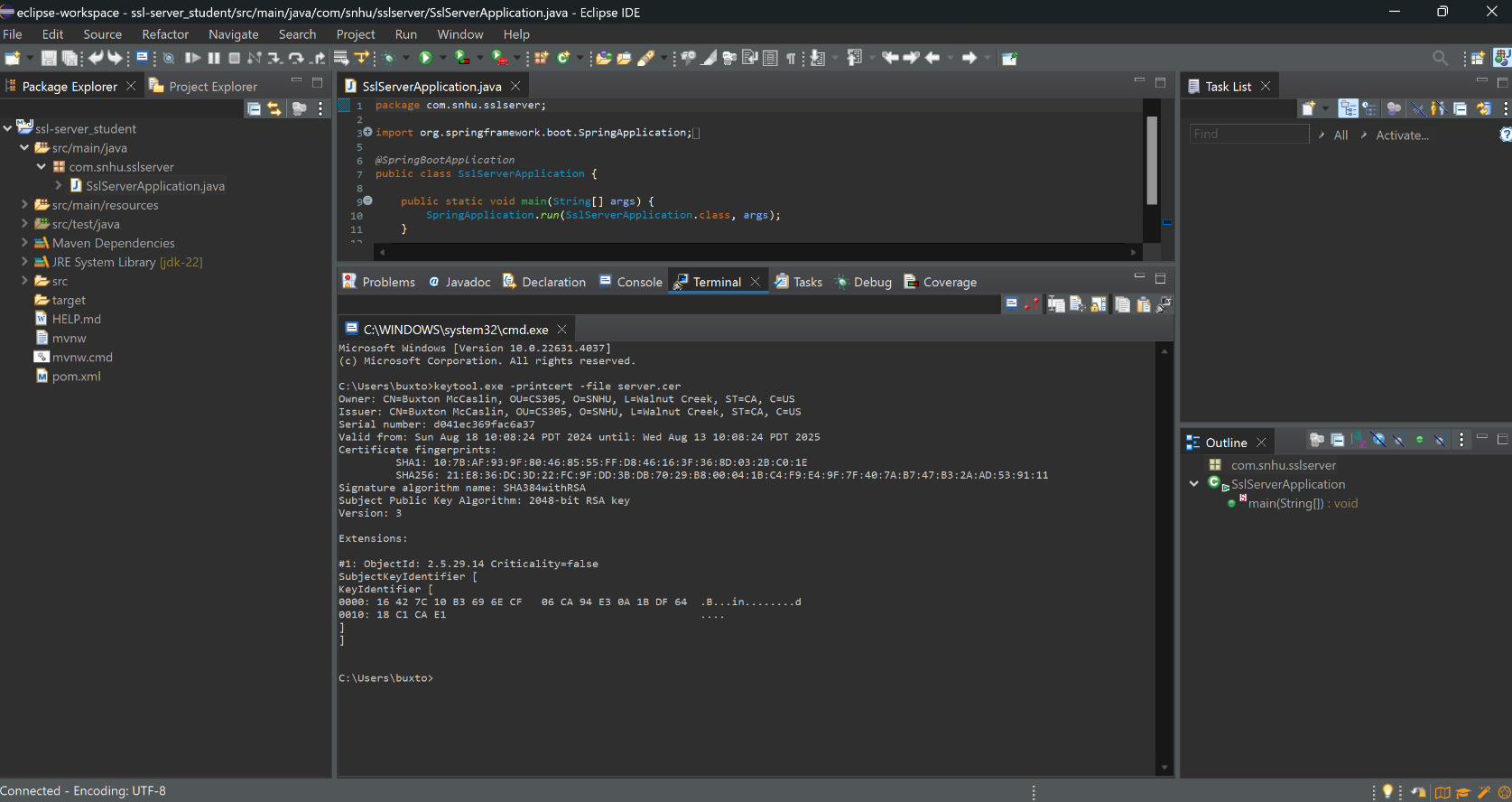
Buxton McCaslin

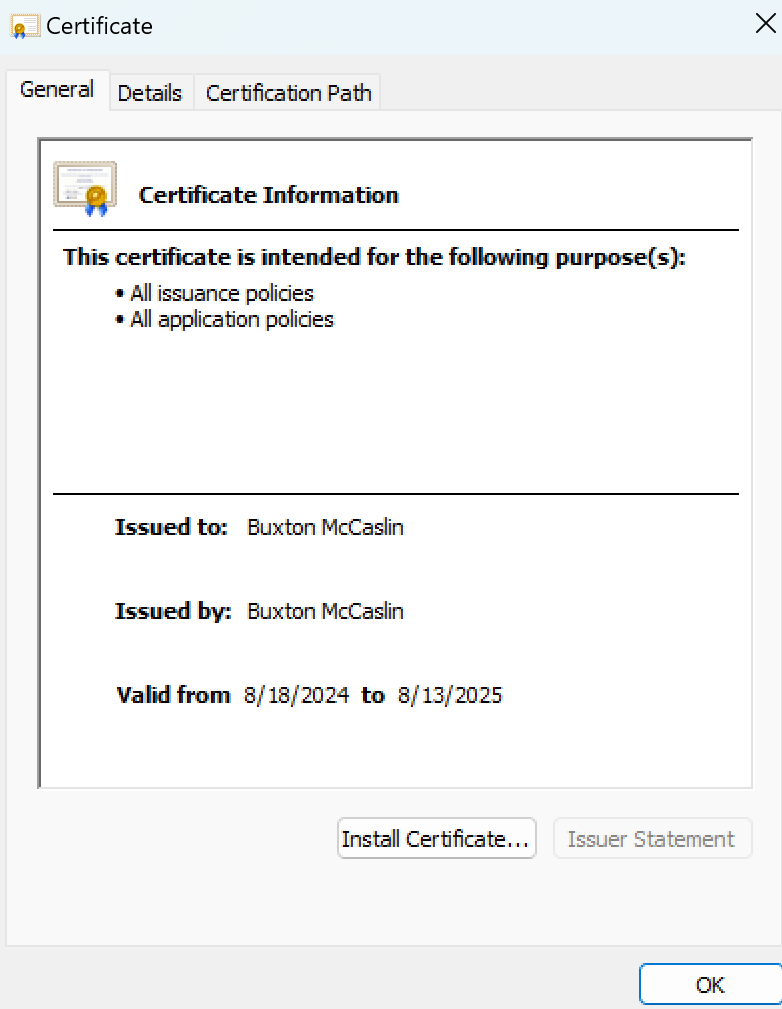
## Algorithm Cipher

* 1. *Provide a brief, high-level overview of the encryption algorithm cipher.*
     1. Artemis Financial’s primary objective is to provide financial services internationally and in consideration of protecting sensitive information, I would recommend SHA-256 as the most appropriate encryption algorithm cipher for the client. SHA-256 is one of the most secure hash functions and is currently impossible to crack. It is standardized and compatible with a variety of platforms and is also commonly used industry wide for digital signatures, certificates, and blockchain technology.
  2. *Discuss the hash functions and bit levels of the cipher.*
     1. The hash functions and bit levels of SHA-256 are made randomly. It operates by taking any length of input data and outputs a 256-bit (32-byte) hash value, which is what makes it highly resistant to cryptographic attacks and collisions. While the bit levels will determine the length of the encryption. It is also non-reversible, meaning it is not possible to backtrack from the hash value to the original data, adding an additional layer of security. Additionally, with even the smallest change in input data, it can create an entirely different hash value, making it perfect for maintaining a high level of data integrity.
  3. *Explain the use of random numbers, symmetric versus non-symmetric keys, and so on.*
     1. The use of random numbers when it comes to encryption can severely impact the ability of an outside actor to gain unauthorized access as it creates unpredictability. Basically, there is no direct correlation or pattern between values. When comparing symmetric versus asymmetric keys, symmetric keys means that there is only one key used between sender and recipient. The key used to encrypt data is the exact same key to decode. While it is inherently less secure, in some situations clients may benefit from using symmetric keys as it does have faster performance and lower resource consumption in comparison. Whereas asymmetric keys mean there are two keys being used. One key to encrypt and another to decrypt. Often, it is used as public and private key pair. A public key that can be used by anyone to encrypt while the private key is only held by the intended receiver for decryption.
  4. *Describe the history and current state of encryption algorithms.*
     1. One of the first instances of encryption algorithms dates to the 7th century when the ancient Spartans invented a device called the scytale. This technology consisted of a wooden rod with a specific diameter and parchment strips with a message written on them. By wrapping the strips onto the rod, the message could be read. Later on, the Caesar Cipher was created during the Roman Empire and was widely used even as late as the 2000s. Over the centuries, encryption algorithms have evolved from these simple methods to the highly sophisticated systems we use today, such as RSA and AES. These systems are very complex and rely on advanced mathematical concepts and large amounts of computer power to keep data secure. As the world continues to advance and become more digital, the importance of encryption continues to escalate. Humanity has and always will have a strong commitment to prioritizing security and privacy.

## Certificate Generation

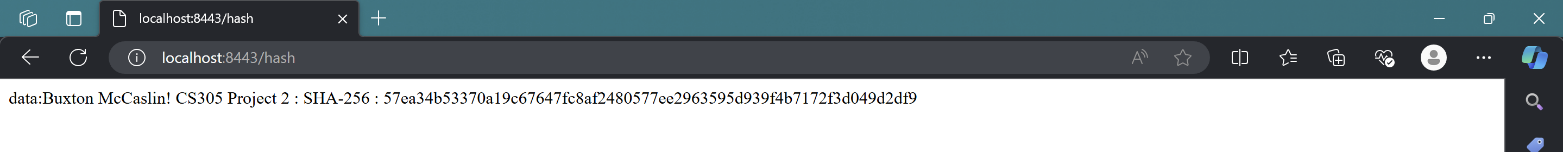
Insert a screenshot below of the CER file.





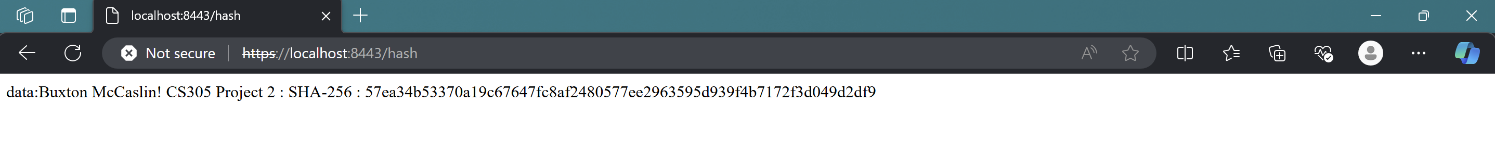
## Deploy Cipher

Insert a screenshot below of the checksum verification.



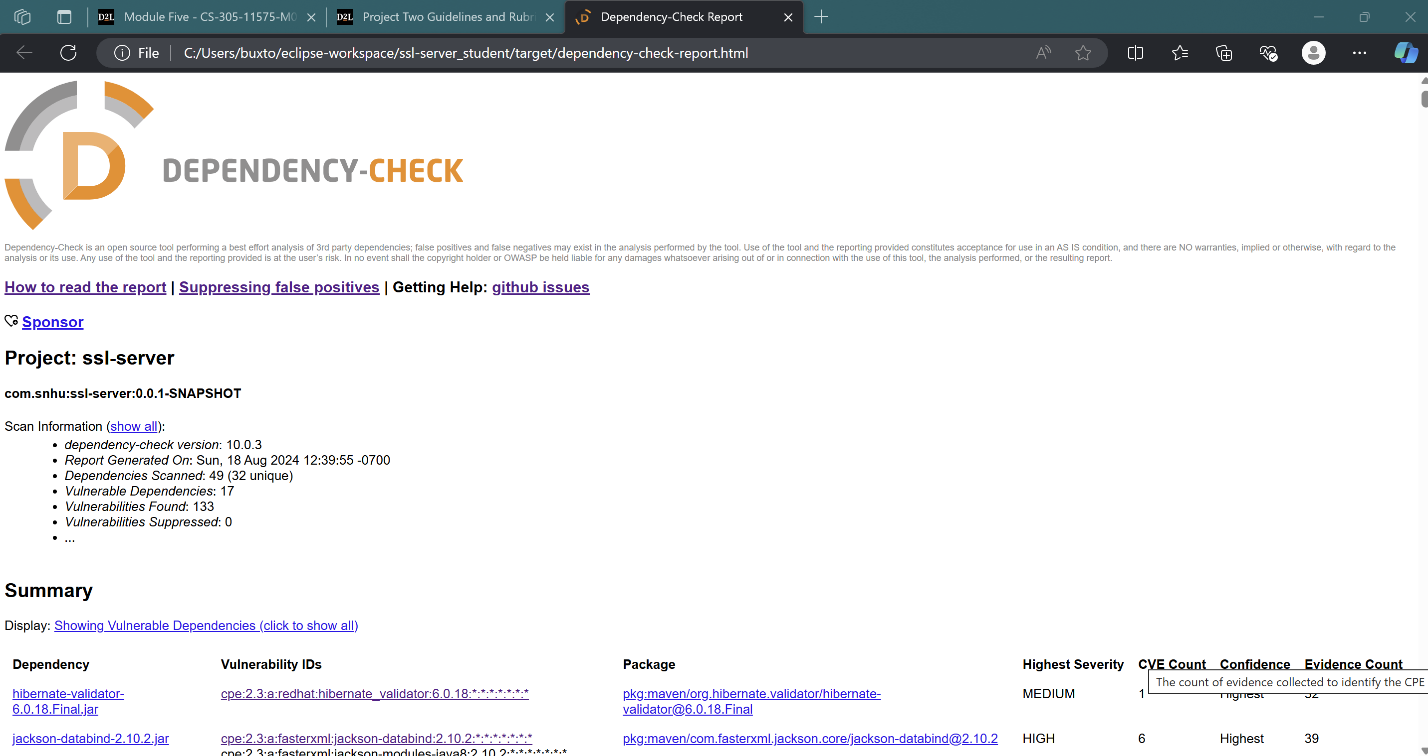
## Secure Communications

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



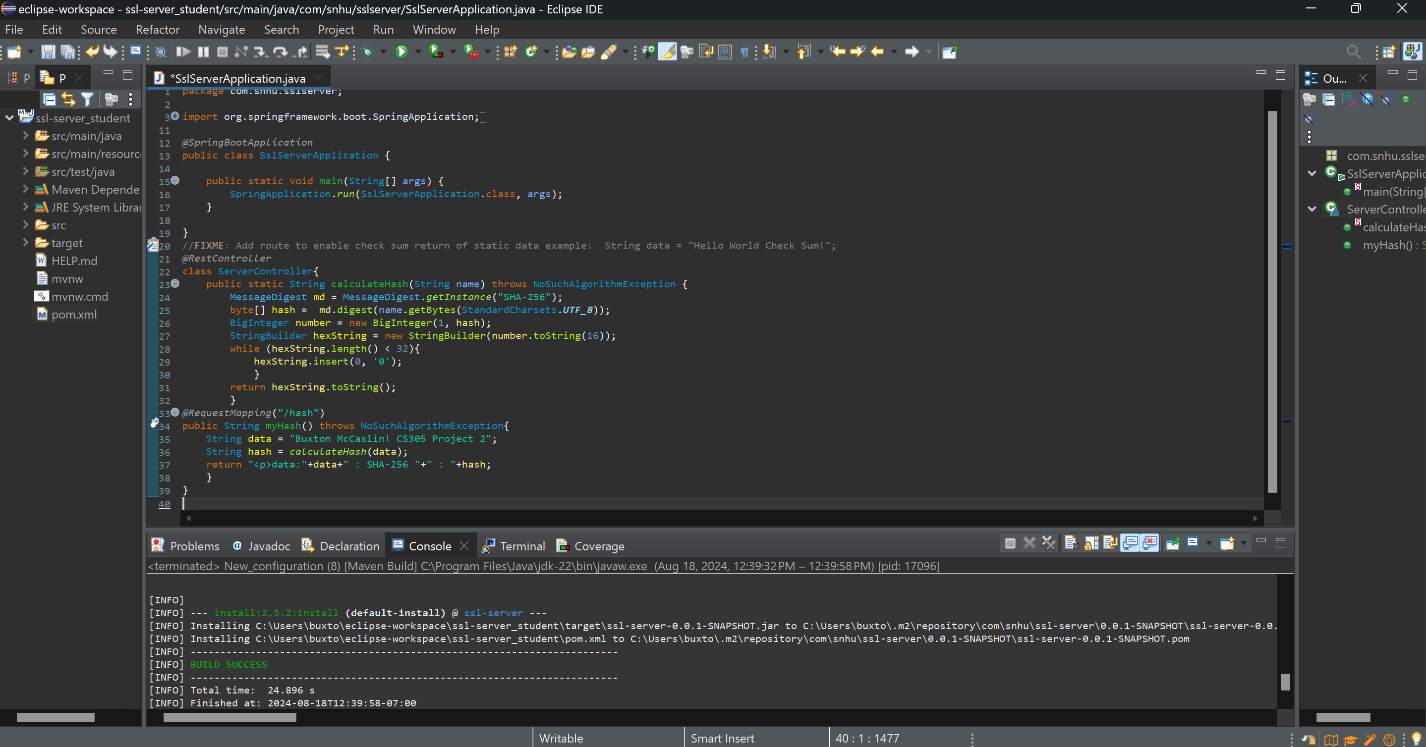
## Secondary Testing

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



## Functional Testing

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



## Summary

* 1. Refer to the vulnerability assessment process flow diagram in the Supporting Materials section. Highlight the areas of security that you addressed by refactoring the code. Discuss your process for adding layers of security to the software application.
     1. The key security enhancement I implemented was the use of self-signed certificates to enable HTTPS for our application. The first step was to ensure these certificates were correctly created, securing our website and instilling confidence in our users that they are interacting with our authentic platform, not an impostor. This measure is essential for maintaining the integrity of our business.

Next, I verified that our hashing function operated as expected by checking it with a checksum. By utilizing the SHA-256 algorithm, known for its effective security and minimal risk of collisions, we can be confident that our customers' data is securely encrypted and difficult to compromise, which significantly strengthens our overall security.

The final step involved patching all vulnerabilities identified during the dependency check in the pom.xml file. This process ensures that our application’s internal components are up to date and functioning as intended. Additionally, I added a controller in the sslServiceApplication.java file, using @RestController and @RequestMapping to guarantee that the program runs correctly when the hash is accessed.

## Industry Standard Best Practices

* 1. Explain how you used industry standard best practices to maintain the software application’s existing security.
     1. One of the ways industry best practices was integrated into the software application’s existing security was through the use of secondary security screenings. Through Maven, a vulnerability scan was conducted that allowed me to revise and patch the code. Keeping the software and applications utilized up to date is one of the best practices for secure coding. Additionally, another best practice that was used with this software was encryption. By enabling a checksum that follows HTTPS protocol for the program, this verifies a secure link for both client and server. Lastly, another best practice used in this program is access control. Through the process of creating a certificate, privileges can be limited to what is necessary for their roles.
  2. Explain the value of applying industry standard best practices for secure coding to the company’s overall well-being.
     1. There is an overabundance of value in applying industry standard best practices for secure coding. Only by applying these practices can programmers mitigate penetration and attacks before they happen. It is essential to maintaining not only the integrity of the software, but also the well-being of the company. In consideration of the privacy and sensitive data that Artemis Financial would be working with, it is extremely important to protect and provide an effective defense for protecting the system.

## References

Crane, C. (2020, November 4). *Symmetric Encryption 101: Definition, How It Works & When It’s Used*. Hashed out by the SSL StoreTM. <https://www.thesslstore.com/blog/symmetric-encryption-101-definition-how-it-works-when-its-used/>

Harvey, S. (2020, February 19). *8 Secure Coding Best Practices Learned from OWASP*. KirkpatrickPrice Home. <https://kirkpatrickprice.com/blog/secure-coding-best-practices/>

Schneider, J. (2024, July 17). *The History of Cryptography | IBM*. Ibm.com. <https://www.ibm.com/think/topics/cryptography-history>

Selinger, B. (2023, October 15). *Encryption: A Timeline of How It Changed the World | ExpressVPN Blog*. Home of Internet Privacy. <https://www.expressvpn.com/blog/encryption-history/>