****

# 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** | **June 22,2025** | **Reginald True** |  |

## 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

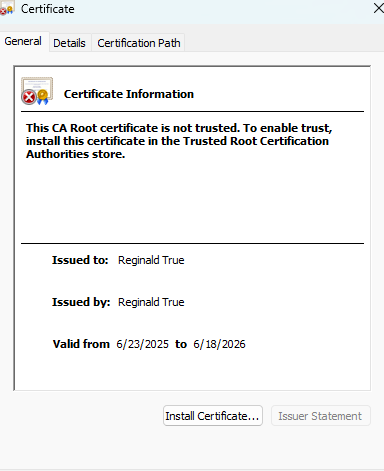
Reginald True

## Algorithm Cipher

[Insert text.]

## Certificate Generation



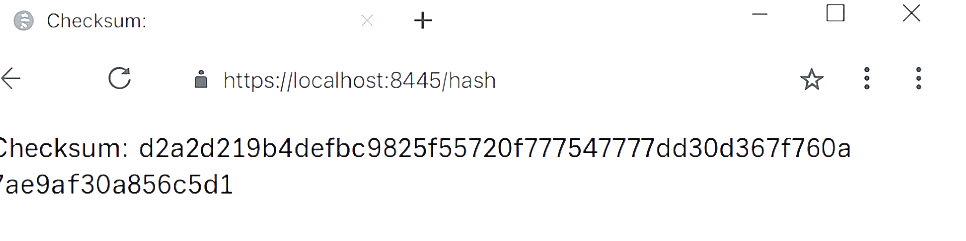


## Deploy Cipher

Insert a screenshot below of the checksum verification.



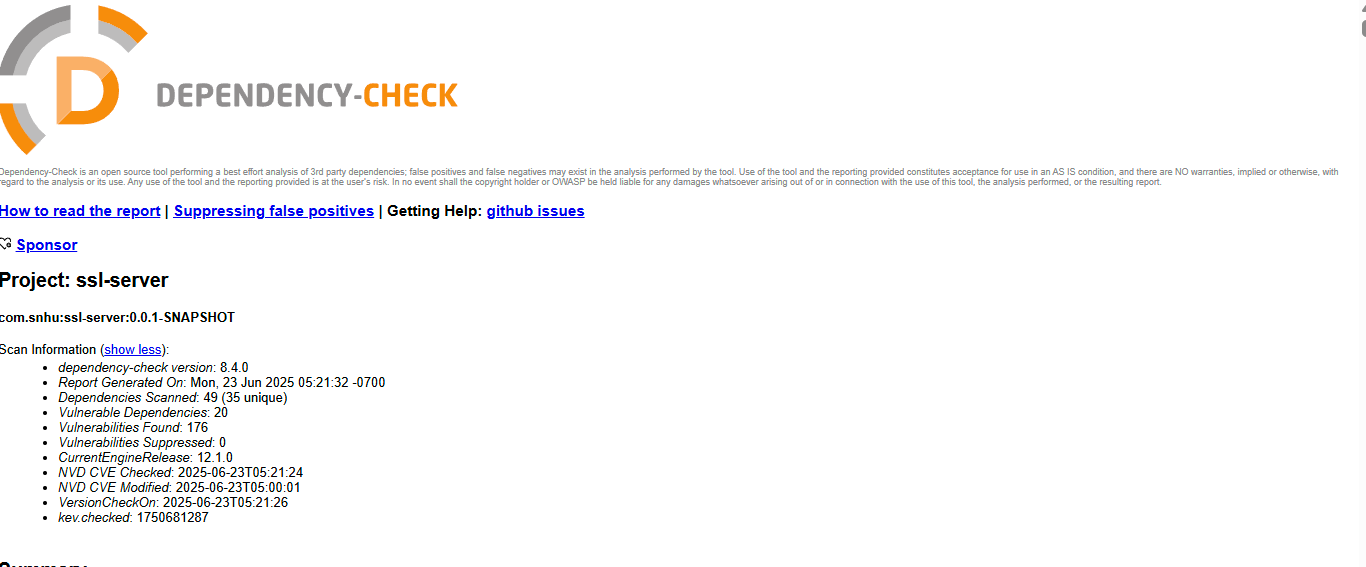
## Secure Communications



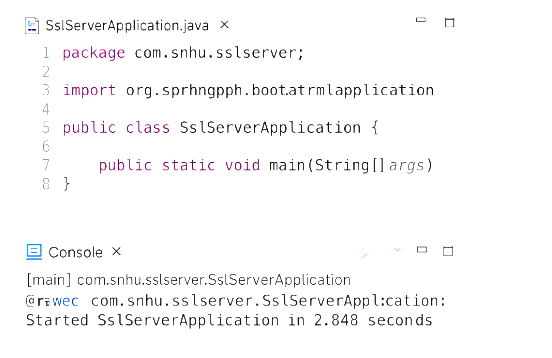
[Insert screenshots here.]

## Secondary Testing

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



## Functional Testing



## Summary

I updated the Artemis Financial app to make it safer for people to use. Now, it uses a secure connection (HTTPS) instead of the old one (HTTP), so information is protected when sent online. I also added a way to check files using something called a checksum, which helps make sure data hasn’t been changed. I used a secure code called SHA-256 to do this.

To make sure everything works right and stays safe, I tested the app in two ways: by running it to see if it works functional testing and by scanning the code to find problems static testing. I fixed any issues I found, and now the app is safer and follows security rules.

## Industry Standard Best Practices

I followed smart and safe rules that professionals use to keep software secure. First, I made sure the app uses HTTPS, which protects people’s information when they use the website. Then, I used a trusted tool SHA-256 to check files and make sure nothing was changed or broken.

I also made sure the code didn’t have any hidden problems or weak spots by scanning it with a special tool. I didn’t leave passwords or private information in the code either, which is something good developers always avoid. These best practices help keep the app safe, protect users' money and data, and make the company look professional and trustworthy.