****

# 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 18, 2025** | **Emerald Tresch** |  |

## 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

Emerald Tresch

## Algorithm Cipher

[Insert text.]

## Certificate Generation

Insert a screenshot below of the CER file.

[Insert screenshots here.]

## Deploy Cipher

Insert a screenshot below of the checksum verification.

[Insert screenshots here.]

## Secure Communications

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

[Insert screenshots here.]

## Secondary Testing

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

[Insert screenshots here.]

## Functional Testing

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

[Insert screenshots here.]

## Summary

[Insert text.]

## Industry Standard Best Practices

[Insert text.]

2025-06-19 20:50:33.244 INFO 40176 --- [ main] o.s.web.context.ContextLoader : Root WebApplicationContext: initialization completed in 866 ms

2025-06-19 20:50:33.564 INFO 40176 --- [ main] o.s.s.concurrent.ThreadPoolTaskExecutor : Initializing ExecutorService 'applicationTaskExecutor'

2025-06-19 20:50:34.144 INFO 40176 --- [ main] o.apache.catalina.core.StandardService : Stopping service [Tomcat]

2025-06-19 20:50:34.148 INFO 40176 --- [ main] ConditionEvaluationReportLoggingListener :

Error starting ApplicationContext. To display the conditions report re-run your application with 'debug' enabled.

2025-06-19 20:50:34.152 ERROR 40176 --- [ main] o.s.boot.SpringApplication : Application run failed

Diagnosing the Error in ApplicationContext

It seems that the issue lies in the initialization of the ApplicationContext, as indicated by the error message: "Error starting ApplicationContext." This suggests that there may be a misconfiguration or unmet condition in your application's setup. Below are potential reasons for this error and steps to address it:

# Possible Causes

* Configuration Issues: There may be errors or omissions in your application properties or configuration files, such as `application.yml` or `application.properties`.
* Dependency Conflicts: One or more dependencies in your project may have compatibility issues or missing versions.
* Bean Initialization Failure: A Spring bean might not be correctly defined or initialized, leading to this failure.
* Unmet Conditions: Certain configuration conditions, such as profiles or environment settings, may not have been satisfied.

# Steps to Resolve

* Enable Debugging: Re-run the application with the `--debug` flag to get a detailed condition evaluation report. This will pinpoint the specific cause of the error.
* Check Logs: Examine the detailed logs preceding the error message for clues about what went wrong.
* Verify Dependencies: Ensure that all dependencies in your project are up-to-date and compatible with your Spring Boot version.
* Validate Configuration: Double-check your configuration files for typos, missing values, or incorrect profiles.
* Review Beans: Inspect your Spring beans for any missing annotations, such as `@Component` or `@Service`, and ensure they are correctly wired.
* Run Tests: Perform unit and integration testing on the application to isolate the issue to a specific component or module.

By following these steps, you should be able to identify and resolve the problem causing the ApplicationContext error. If the issue persists, consider seeking help from a community forum or consulting the Spring Boot documentation.