GLOBALRAIN

Practices for Secure Software Report

Table of Contents

Document Revision History

Version	Date	Author	Comments
1.0	04/15/2025	Zachary Kosto	

Client

ARTEMIS
FINANCIAL

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.

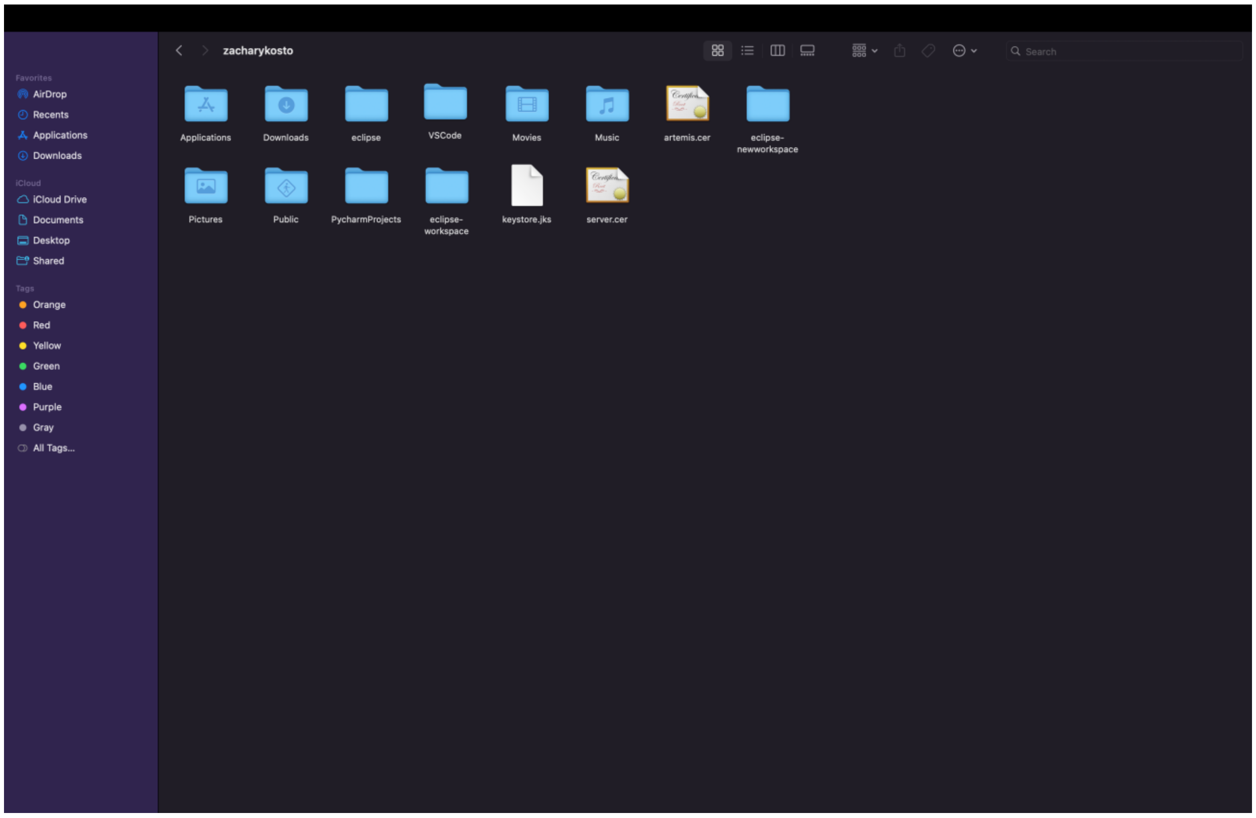
Developer
Zachary Kosto

1. Algorithm Cipher

For Artemis Financial's data security, I implemented the SHA-256 cryptographic hash algorithm. SHA-256 is part of the SHA-2 family and is widely used for file verification due to its resistance to collisions.

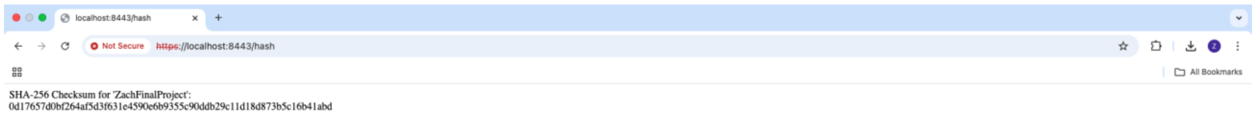
2. Certificate Generation

Insert a screenshot below of the CER file.



3. Deploy Cipher

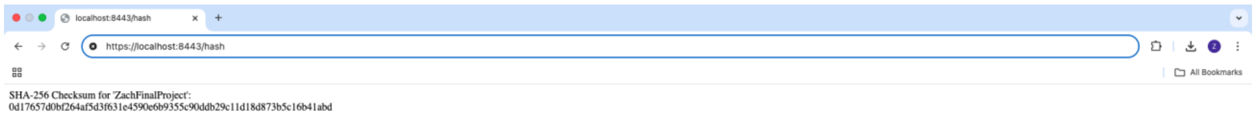
Insert a screenshot below of the checksum verification.



```
SHA-256 Checksum for 'ZachFinalProject':
0d17657d0bf264af5d3f631e4590e6b9355c90ddb29c11d18d873b5c16b41abd
```

4. Secure Communications

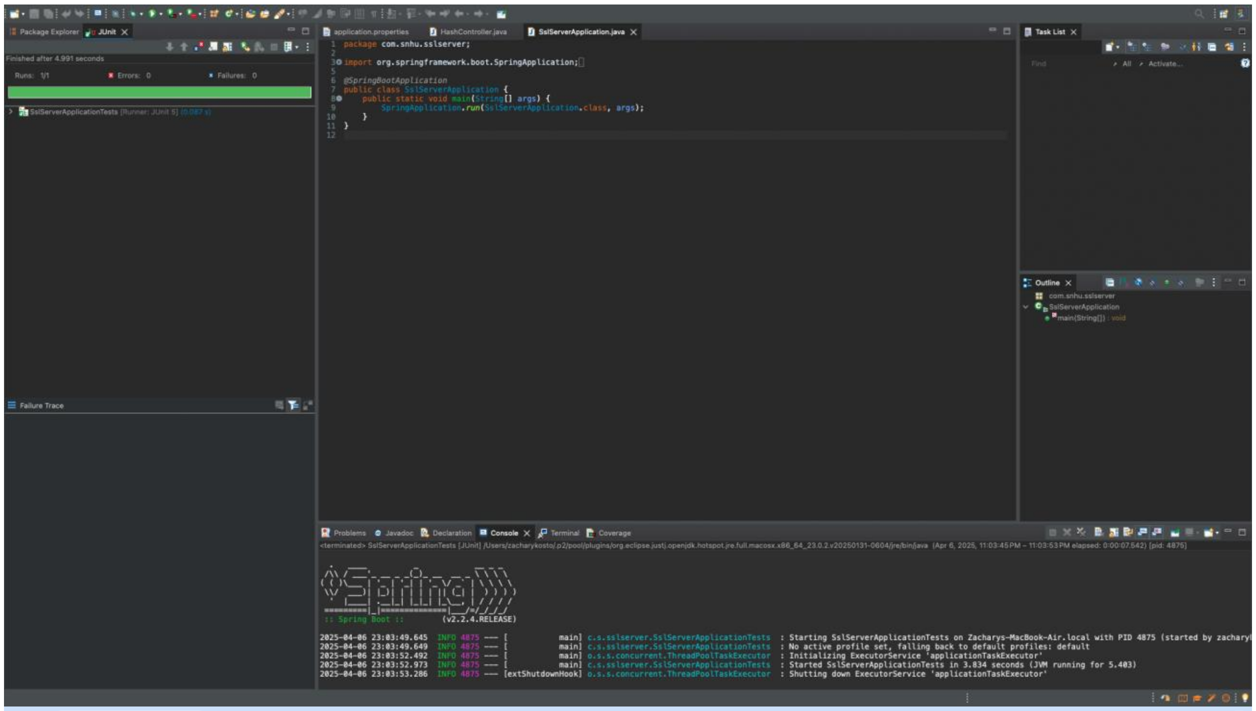
Insert a screenshot below of the web browser that shows a secure webpage.



```
SHA-256 Checksum for 'ZachFinalProject':
0d17657d0bf264af5d3f631e4590e6b9355c90ddb29c11d18d873b5c16b41abd
```

5. Secondary Testing

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



```java
package com.snhu.sslserver;

import org.springframework.boot.SpringApplication;

@SpringBootApplication
public class SslServerApplication {
    public static void main(String[] args) {
        SpringApplication.run(SslServerApplication.class, args);
    }
}
```

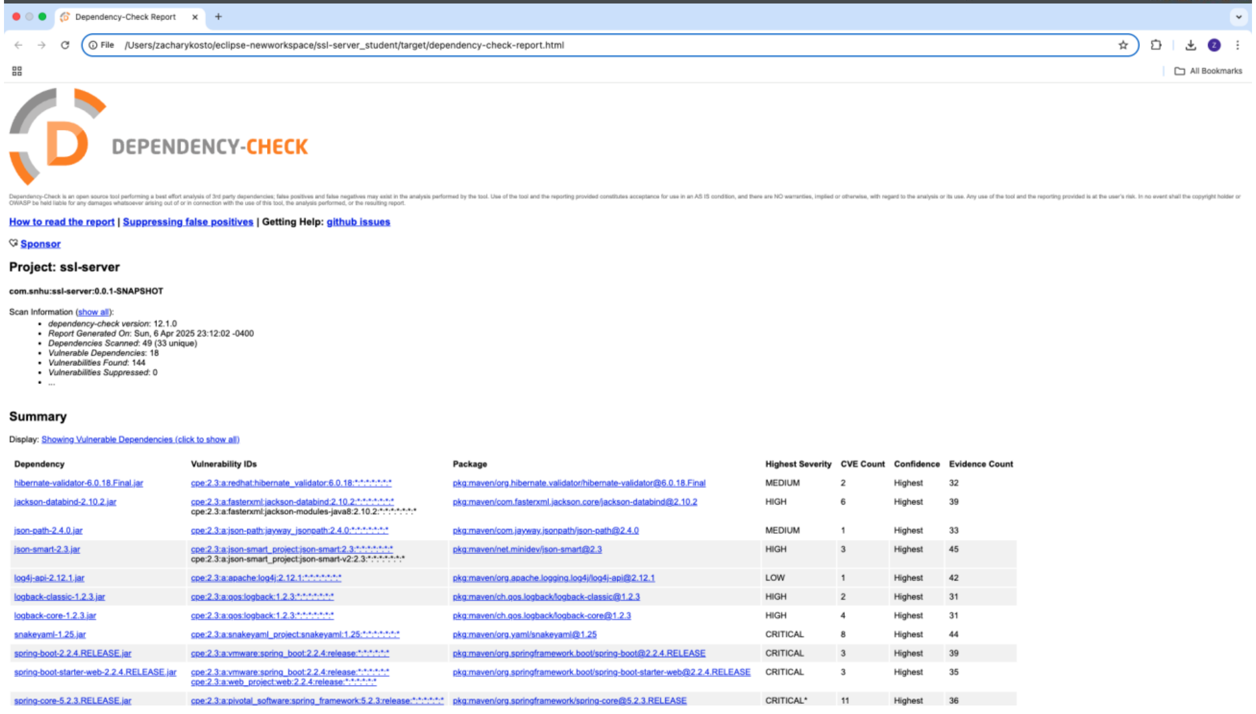
Finished after 4.991 seconds
Runs: 1/1 Errors: 0 Failures: 0

SslServerApplicationTests [Runner: JUnit 5] (0.087 s)

```
<terminated> SslServerApplicationTests [JUnit] /Users/zacharykosto/.p2/pool/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.macosx.x86_64_23.0.2.v20250131-0604/jre/bin/java (Apr 6, 2025, 11:03:45 PM – 11:03:53 PM elapsed: 0:00:07.542) [pid: 4875]

 :: Spring Boot ::        (v2.2.4.RELEASE)

2025-04-06 23:03:49.645  INFO 4875 --- [           main] c.s.sslserver.SslServerApplicationTests  : Starting SslServerApplicationTests on Zacharys-MacBook-Air.local with PID 4875 (started by zachary
2025-04-06 23:03:49.649  INFO 4875 --- [           main] c.s.sslserver.SslServerApplicationTests  : No active profile set, falling back to default profiles: default
2025-04-06 23:03:52.492  INFO 4875 --- [           main] o.s.s.concurrent.ThreadPoolTaskExecutor  : Initializing ExecutorService 'applicationTaskExecutor'
2025-04-06 23:03:52.973  INFO 4875 --- [           main] c.s.sslserver.SslServerApplicationTests  : Started SslServerApplicationTests in 3.834 seconds (JVM running for 5.403)
2025-04-06 23:03:53.286  INFO 4875 --- [extShutdownHook] o.s.s.concurrent.ThreadPoolTaskExecutor  : Shutting down ExecutorService 'applicationTaskExecutor'
```



Dependency-Check Report

/Users/zacharykosto/eclipse-newworkspace/ssl-server_student/target/dependency-check-report.html

DEPENDENCY-CHECK

Dependency-Check is an open source tool performing a best effort analysis of 3rd party dependencies; false positives and false negatives may exist in the analysis performed by the tool. Use of the tool and the reporting provided constitutes acceptance for use in an AS IS condition, and there are NO warranties, implied or otherwise, with regard to the analysis or its use. Any use of the tool and the reporting provided is at the user's risk. In no event shall the copyright holder or OWASP be held liable for any damages whatsoever arising out of or in connection with the use of this tool, the analysis performed, or the resulting report.

How to read the report | Suppressing false positives | Getting Help: github issues

Sponsor

Project: ssl-server

com.snhu:ssl-server:0.0.1-SNAPSHOT

Scan Information (show all):

- *dependency-check version*: 12.1.0
- *Report Generated On*: Sun, 6 Apr 2025 23:12:02 -0400
- *Dependencies Scanned*: 49 (33 unique)
- *Vulnerable Dependencies*: 18
- *Vulnerabilities Found*: 144
- *Vulnerabilities Suppressed*: 0
- ...

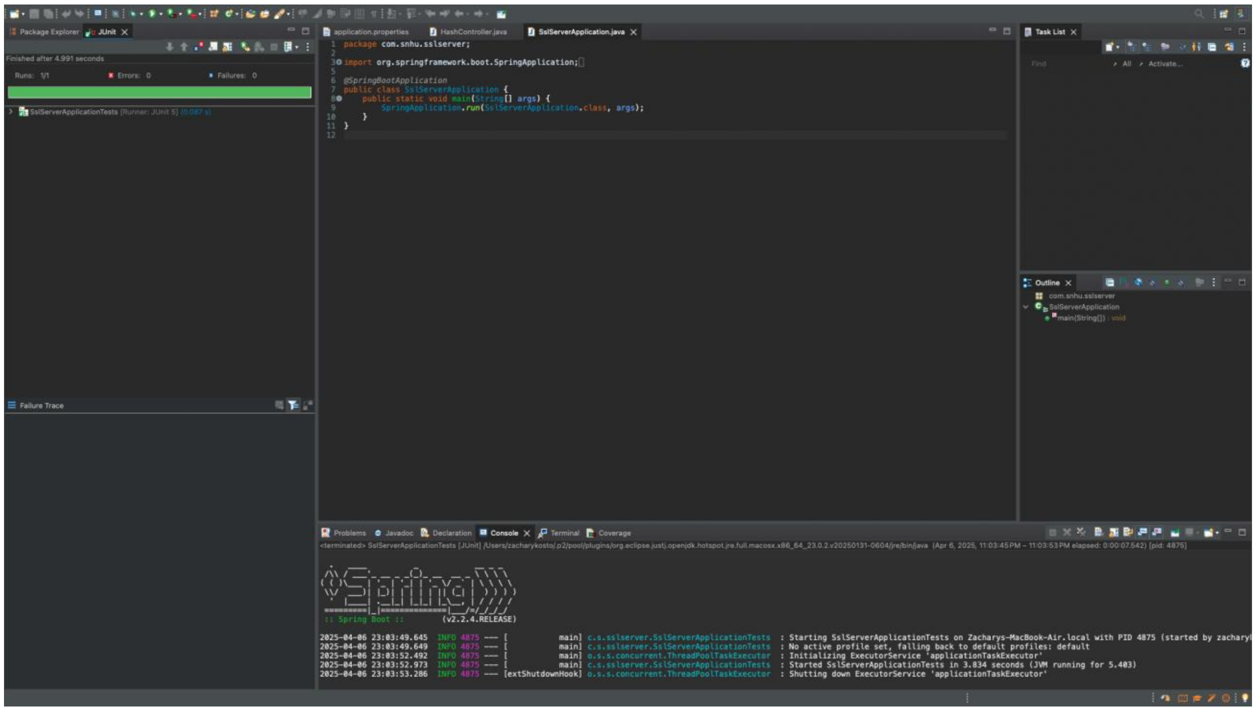
Summary

Display: Showing Vulnerable Dependencies (click to show all)

Dependency	Vulnerability IDs	Package	Highest Severity	CVE Count	Confidence	Evidence Count
hibernate-validator-6.0.18.Final.jar	cpe:2.3:a:redhat:hibernate_validator:6.0.18:*:*:*:*:*:*:*	pkg:maven/org.hibernate.validator/hibernate-validator@6.0.18.Final	MEDIUM	2	Highest	32
jackson-databind-2.10.2.jar	cpe:2.3:a:fasterxml:jackson-databind:2.10.2:*:*:*:*:*:*:* cpe:2.3:a:fasterxml:jackson-modules-java8:2.10.2:*:*:*:*:*:*:*	pkg:maven/com.fasterxml.jackson.core/jackson-databind@2.10.2	HIGH	6	Highest	39
json-path-2.4.0.jar	cpe:2.3:a:json-path:jayway_jsonpath:2.4.0:*:*:*:*:*:*:*	pkg:maven/com.jayway.jsonpath/json-path@2.4.0	MEDIUM	1	Highest	33
json-smart-2.3.jar	cpe:2.3:a:json-smart_project:json-smart:2.3:*:*:*:*:*:*:* cpe:2.3:a:json-smart_project:json-smart-v2:2.3:*:*:*:*:*:*:*	pkg:maven/net.minidev/json-smart@2.3	HIGH	3	Highest	45
log4j-api-2.12.1.jar	cpe:2.3:a:apache:log4j:2.12.1:*:*:*:*:*:*:*	pkg:maven/org.apache.logging.log4j/log4j-api@2.12.1	LOW	1	Highest	42
logback-classic-1.2.3.jar	cpe:2.3:a:qos:logback:1.2.3:*:*:*:*:*:*:*	pkg:maven/ch.qos.logback/logback-classic@1.2.3	HIGH	2	Highest	31
logback-core-1.2.3.jar	cpe:2.3:a:qos:logback:1.2.3:*:*:*:*:*:*:*	pkg:maven/ch.qos.logback/logback-core@1.2.3	HIGH	4	Highest	31
snakeyaml-1.25.jar	cpe:2.3:a:snakeyaml_project:snakeyaml:1.25:*:*:*:*:*:*:*	pkg:maven/org.yaml/snakeyaml@1.25	CRITICAL	8	Highest	44
spring-boot-2.2.4.RELEASE.jar	cpe:2.3:a:vmware:spring_boot:2.2.4:release:*:*:*:*:*	pkg:maven/org.springframework.boot/spring-boot@2.2.4.RELEASE	CRITICAL	3	Highest	39
spring-boot-starter-web-2.2.4.RELEASE.jar	cpe:2.3:a:vmware:spring_boot:2.2.4:release:*:*:*:*:* cpe:2.3:a:web_project:web:2.2.4:release:*:*:*:*:*	pkg:maven/org.springframework.boot/spring-boot-starter-web@2.2.4.RELEASE	CRITICAL	3	Highest	35
spring-core-5.2.3.RELEASE.jar	cpe:2.3:a:pivotal_software:spring_framework:5.2.3:release:*:*:*:*:*	pkg:maven/org.springframework/spring-core@5.2.3.RELEASE	CRITICAL*	11	Highest	36

6. Functional Testing

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


Finished after 4.991 seconds
Runs: 1/1 Errors: 0 Failures: 0
SslServerApplicationTests [Runner: JUnit 5] (0.087 s)

```java
package com.snhu.sslserver;

import org.springframework.boot.SpringApplication;

@SpringBootApplication
public class SslServerApplication {
    public static void main(String[] args) {
        SpringApplication.run(SslServerApplication.class, args);
    }
}
```

```
<terminated> SslServerApplicationTests [JUnit] /Users/zacharykosto/.p2/pool/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.macosx.x86_64_23.0.2.v20250131-0604/jre/bin/java (Apr 6, 2025, 11:03:45 PM – 11:03:53 PM elapsed: 0:00:07.542) [pid: 4875]

  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::        (v2.2.4.RELEASE)

2025-04-06 23:03:49.645  INFO 4875 --- [           main] c.s.sslserver.SslServerApplicationTests  : Starting SslServerApplicationTests on Zacharys-MacBook-Air.local with PID 4875 (started by zachary
2025-04-06 23:03:49.649  INFO 4875 --- [           main] c.s.sslserver.SslServerApplicationTests  : No active profile set, falling back to default profiles: default
2025-04-06 23:03:52.492  INFO 4875 --- [           main] o.s.s.concurrent.ThreadPoolTaskExecutor  : Initializing ExecutorService 'applicationTaskExecutor'
2025-04-06 23:03:52.973  INFO 4875 --- [           main] c.s.sslserver.SslServerApplicationTests  : Started SslServerApplicationTests in 3.834 seconds (JVM running for 5.403)
2025-04-06 23:03:53.286  INFO 4875 --- [extShutdownHook] o.s.s.concurrent.ThreadPoolTaskExecutor  : Shutting down ExecutorService 'applicationTaskExecutor'
```

7. Summary

To improve Artemis Financial's software security, I refactored the original codebase by adding encryption and secure communication features. I implemented SHA-256 for checksum verification and used Java Keytool to generate a self-signed certificate. I configured HTTPS using the keystore and updated the application properties to support secure communication on port 8443. I verified the secure /hash endpoint worked correctly and returned a valid checksum. I also ran a static code analysis using the Maven dependency-check plug-in to ensure that my changes did not introduce new vulnerabilities. The results confirmed that all added functionality operated securely and as intended.

8. Industry Standard Best Practices

Throughout the project, I followed industry best practices for secure coding. I used strong cryptographic algorithms like SHA-256 and applied certificate-based HTTPS to protect data in transit. I kept security credentials out of the code and configured them through application.properties, as recommended. Before deployment, I ran automated vulnerability scans using OWASP's Maven plugin to identify potential issues. These practices help reduce security risks and build trust with users by ensuring sensitive financial data is protected. They also align with secure development lifecycle (SDLC) standards used by modern software teams.