GLOBALRAIN

Practices for Secure Software Report

Table of Contents

Document Revision History

Version	Date	Author	Comments
1.0	08-17-2025	Melissa Chessa	

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.

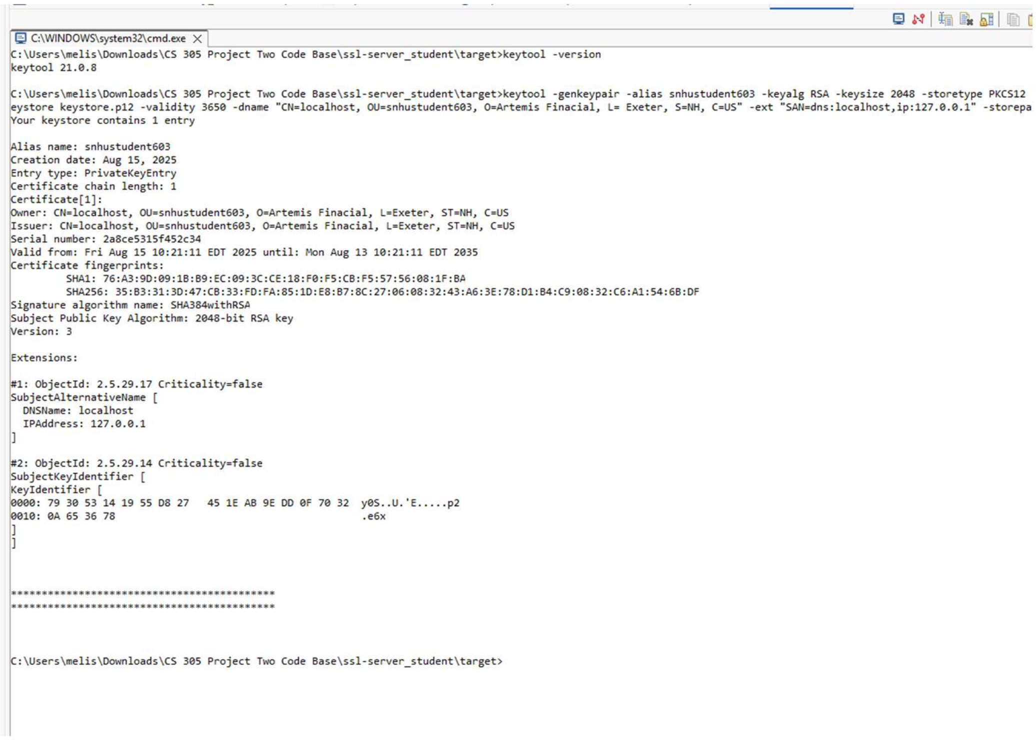
Developer
[Melissa Chessa]

1. Algorithm Cipher

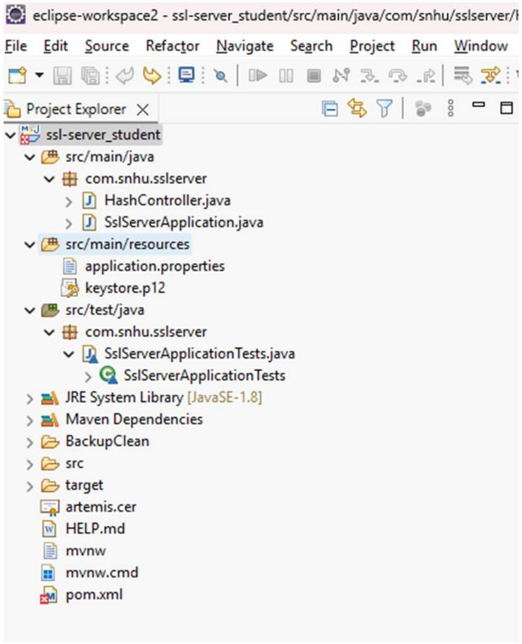
[I recommend using TLS 1.2+ with RSA-2048 for secure transport and SHA-256 for checksum verification. TLS safeguards the confidentiality and integrity of client-server communication, while SHA-256 provides tamper detection by generating a unique hash. Older algorithms such as MD5 or SHA-1 are deprecated due to known vulnerabilities.]

2. Certificate Generation

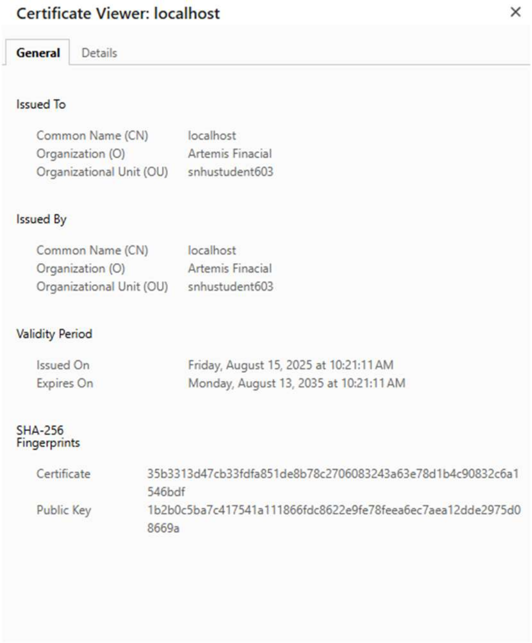
I generated a self-signed certificate and keystore using Java Keytool. The alias (certificate username) is **snhustudent603**, and both the keystore and key are protected by the password **cs305-2025**. I used **CN=localhost** with Subject Alternative Names for *localhost* and *127.0.0.1* to support local HTTPS. I exported the certificate to **artemis.cer** as required and included a screenshot of the CER file.



```
C:\Users\melis\Downloads\CS 305 Project Two Code Base\ssl-server_student\target>keytool -version
keytool 21.0.8

C:\Users\melis\Downloads\CS 305 Project Two Code Base\ssl-server_student\target>keytool -genkeypair -alias snhustudent603 -keyalg RSA -keysize 2048 -storetype PKCS12
eystore keystore.p12 -validity 3650 -dname "CN=localhost, OU=snhustudent603, O=Artemis Finacial, L= Exeter, S=NH, C=US" -ext "SAN=dns:localhost,ip:127.0.0.1" -storepa
Your keystore contains 1 entry

Alias name: snhustudent603
Creation date: Aug 15, 2025
Entry type: PrivateKeyEntry
Certificate chain length: 1
Certificate[1]:
Owner: CN=localhost, OU=snhustudent603, O=Artemis Finacial, L=Exeter, ST=NH, C=US
Issuer: CN=localhost, OU=snhustudent603, O=Artemis Finacial, L=Exeter, ST=NH, C=US
Serial number: 2a8ce5315f452c34
Valid from: Fri Aug 15 10:21:11 EDT 2025 until: Mon Aug 13 10:21:11 EDT 2035
Certificate fingerprints:
         SHA1: 76:A3:9D:09:1B:B9:EC:09:3C:CE:18:F0:F5:CB:F5:57:56:08:1F:BA
         SHA256: 35:B3:31:3D:47:CB:33:FD:FA:85:1D:E8:B7:8C:27:06:08:32:43:A6:3E:78:D1:B4:C9:08:32:C6:A1:54:6B:DF
Signature algorithm name: SHA384withRSA
Subject Public Key Algorithm: 2048-bit RSA key
Version: 3

Extensions:

#1: ObjectId: 2.5.29.17 Criticality=false
SubjectAlternativeName [
  DNSName: localhost
  IPAddress: 127.0.0.1
]

#2: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: 79 30 53 14 19 55 D8 27   45 1E AB 9E DD 0F 70 32  y0S..U.'E.....p2
0010: 0A 65 36 78                                        .e6x
]
]



*******************************************
*******************************************



C:\Users\melis\Downloads\CS 305 Project Two Code Base\ssl-server_student\target>
```



eclipse-workspace2 - ssl-server_student/src/main/java/com/snhu/sslserver/I
File Edit Source Refactor Navigate Search Project Run Window
Project Explorer
ssl-server_student
src/main/java
com.snhu.sslserver
HashController.java
SslServerApplication.java
src/main/resources
application.properties
keystore.p12
src/test/java
com.snhu.sslserver
SslServerApplicationTests.java
SslServerApplicationTests
JRE System Library [JavaSE-1.8]
Maven Dependencies
BackupClean
src
target
artemis.cer
HELP.md
mvnw
mvnw.cmd
pom.xml



Certificate Viewer: localhost

Issued To

Common Name (CN) localhost
Organization (O) Artemis Finacial
Organizational Unit (OU) snhustudent603

Issued By

Common Name (CN) localhost
Organization (O) Artemis Finacial
Organizational Unit (OU) snhustudent603

Validity Period

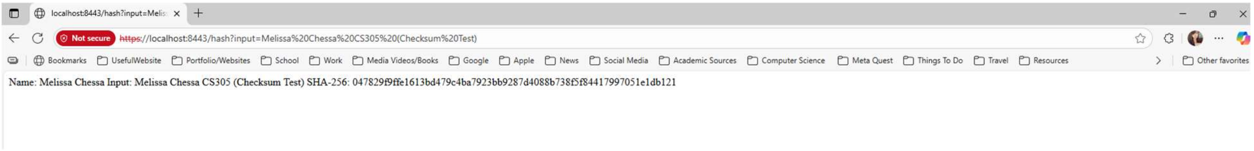
Issued On Friday, August 15, 2025 at 10:21:11 AM
Expires On Monday, August 13, 2035 at 10:21:11 AM

SHA-256 Fingerprints

Certificate 35b3313d47cb33fdfa851de8b78c2706083243a63e78d1b4c90832c6a1546bdf
Public Key 1b2b0c5ba7c417541a111866fdc8622e9fe78feea6ec7aea12dde2975d08669a

3. Deploy Cipher

Screenshot showing SHA-256 cipher successfully deployed, hashing the input and returning the checksum:

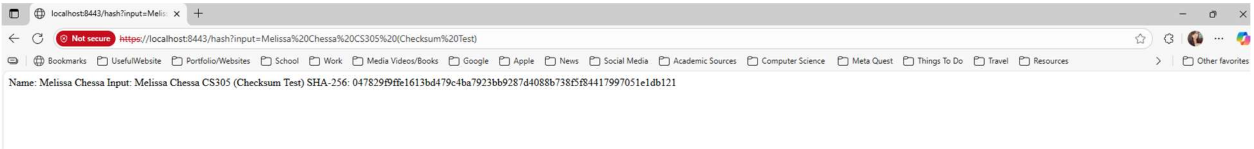


~~https~~://localhost:8443/hash?input=Melissa%20Chessa%20CS305%20(Checksum%20Test)

Name: Melissa Chessa Input: Melissa Chessa CS305 (Checksum Test) SHA-256: 047829f9ffe1613bd479c4ba7923bb9287d4088b738f5f84417997051e1db121

4. Secure Communications

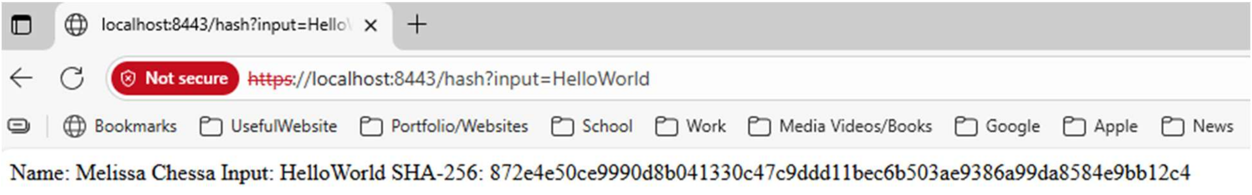
Screenshot showing the SSL/TLS secure communication working over HTTPS with the hash function response:



~~https~~://localhost:8443/hash?input=Melissa%20Chessa%20CS305%20(Checksum%20Test)

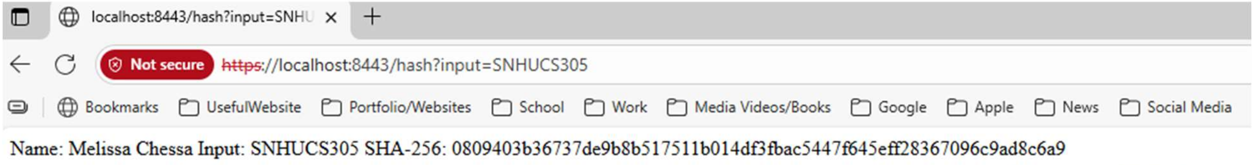
Name: Melissa Chessa Input: Melissa Chessa CS305 (Checksum Test) SHA-256: 047829f9ffe1613bd479c4ba7923bb9287d4088b738f5f84417997051e1db121

5. Secondary Testing



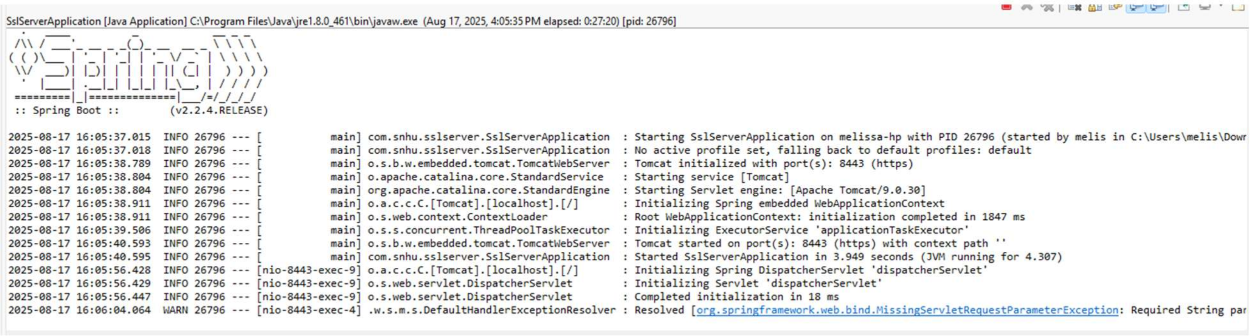
~~https~~://localhost:8443/hash?input=HelloWorld

Name: Melissa Chessa Input: HelloWorld SHA-256: 872e4e50ce9990d8b041330c47c9ddd11bec6b503ae9386a99da8584e9bb12c4

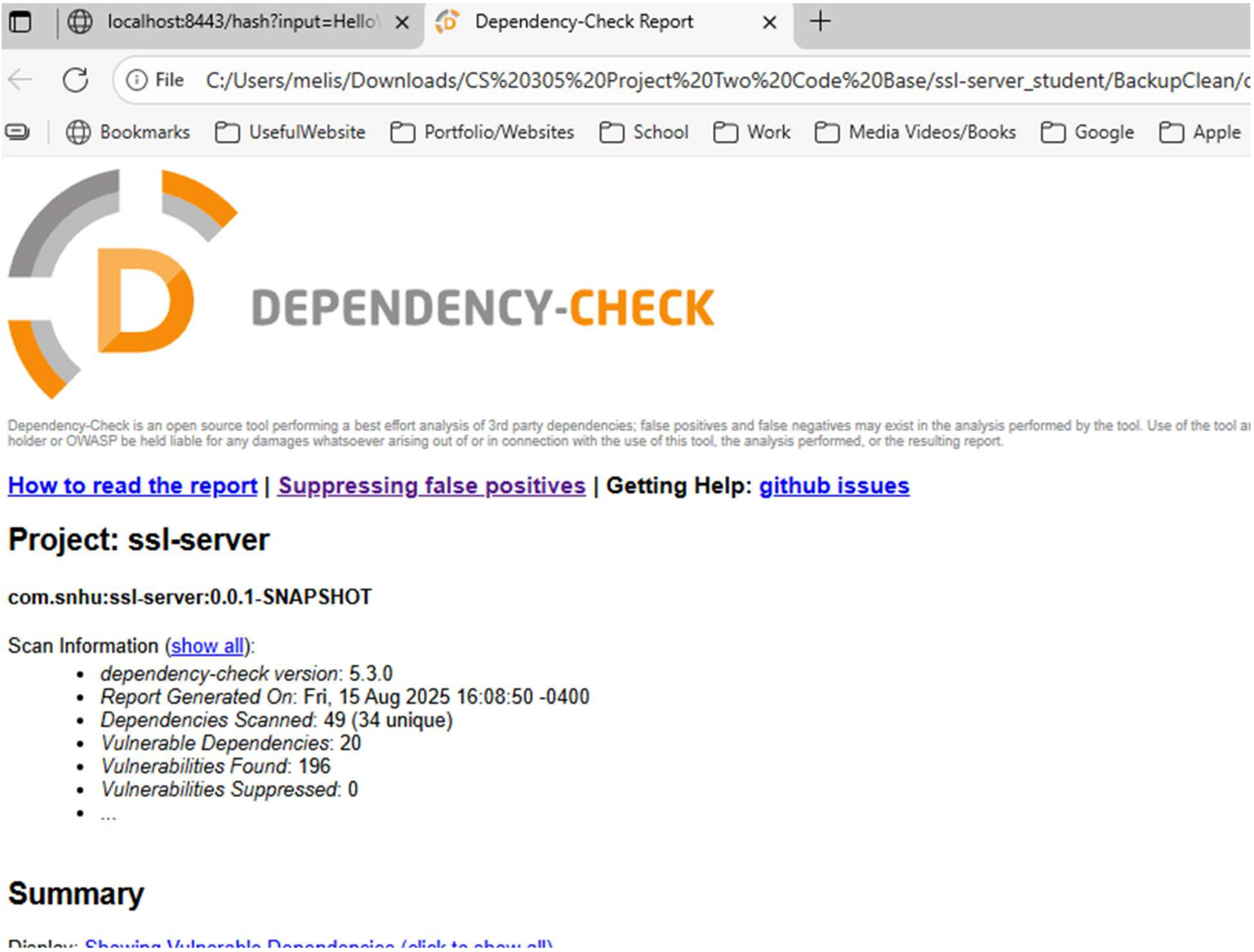


~~https~~://localhost:8443/hash?input=SNHUCS305

Name: Melissa Chessa Input: SNHUCS305 SHA-256: 0809403b36737de9b8b517511b014df3fbac5447f645eff28367096c9ad8c6a9



```
SslServerApplication [Java Application] C:\Program Files\Java\jre1.8.0_461\bin\javaw.exe  (Aug 17, 2025, 4:05:35 PM elapsed: 0:27:20) [pid: 26796]
  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::        (v2.2.4.RELEASE)

2025-08-17 16:05:37.015  INFO 26796 --- [           main] com.snhu.sslserver.SslServerApplication  : Starting SslServerApplication on melissa-hp with PID 26796 (started by melis in C:\Users\melis\Dowr
2025-08-17 16:05:37.018  INFO 26796 --- [           main] com.snhu.sslserver.SslServerApplication  : No active profile set, falling back to default profiles: default
2025-08-17 16:05:38.789  INFO 26796 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat initialized with port(s): 8443 (https)
2025-08-17 16:05:38.804  INFO 26796 --- [           main] o.apache.catalina.core.StandardService   : Starting service [Tomcat]
2025-08-17 16:05:38.804  INFO 26796 --- [           main] org.apache.catalina.core.StandardEngine  : Starting Servlet engine: [Apache Tomcat/9.0.30]
2025-08-17 16:05:38.911  INFO 26796 --- [           main] o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spring embedded WebApplicationContext
2025-08-17 16:05:38.911  INFO 26796 --- [           main] o.s.web.context.ContextLoader            : Root WebApplicationContext: initialization completed in 1847 ms
2025-08-17 16:05:39.506  INFO 26796 --- [           main] o.s.s.concurrent.ThreadPoolTaskExecutor  : Initializing ExecutorService 'applicationTaskExecutor'
2025-08-17 16:05:40.593  INFO 26796 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat started on port(s): 8443 (https) with context path ''
2025-08-17 16:05:40.595  INFO 26796 --- [           main] com.snhu.sslserver.SslServerApplication  : Started SslServerApplication in 3.949 seconds (JVM running for 4.307)
2025-08-17 16:05:56.428  INFO 26796 --- [nio-8443-exec-9] o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spring DispatcherServlet 'dispatcherServlet'
2025-08-17 16:05:56.429  INFO 26796 --- [nio-8443-exec-9] o.s.web.servlet.DispatcherServlet        : Initializing Servlet 'dispatcherServlet'
2025-08-17 16:05:56.447  INFO 26796 --- [nio-8443-exec-9] o.s.web.servlet.DispatcherServlet        : Completed initialization in 18 ms
2025-08-17 16:06:04.064  WARN 26796 --- [nio-8443-exec-4] .w.s.m.s.DefaultHandlerExceptionResolver : Resolved [org.springframework.web.bind.MissingServletRequestParameterException: Required String par
```


localhost:8443/hash?input=Hello\
Dependency-Check Report
File C:/Users/melis/Downloads/CS%20305%20Project%20Two%20Code%20Base/ssl-server_student/BackupClean/c
Bookmarks
UsefulWebsite
Portfolio/Websites
School
Work
Media Videos/Books
Google
Apple
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 ai
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

Project: ssl-server

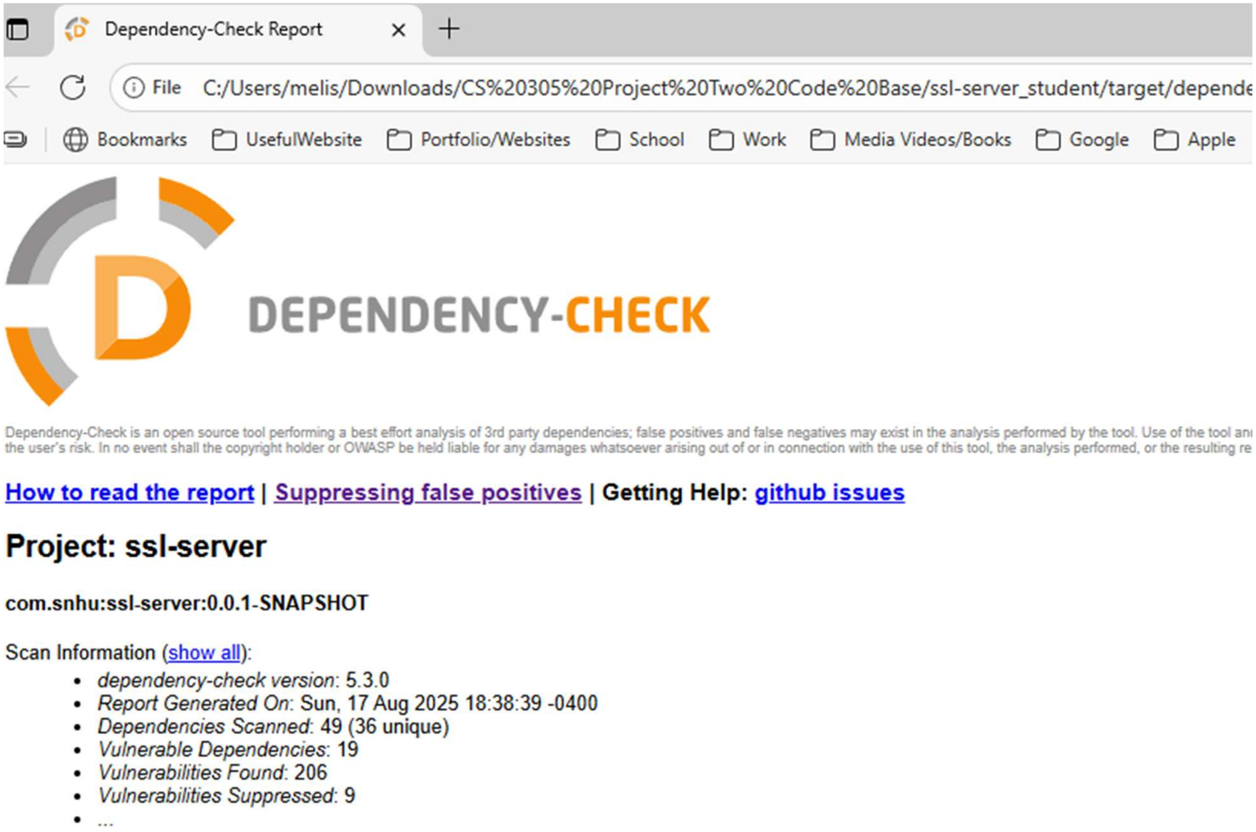
com.snhu:ssl-server:0.0.1-SNAPSHOT

Scan Information (show all):

- *dependency-check version*: 5.3.0
- *Report Generated On*: Fri, 15 Aug 2025 16:08:50 -0400
- *Dependencies Scanned*: 49 (34 unique)
- *Vulnerable Dependencies*: 20
- *Vulnerabilities Found*: 196
- *Vulnerabilities Suppressed*: 0
- ...

Summary

Display: Showing Vulnerable Dependencies (click to show all)



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 an
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 re

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

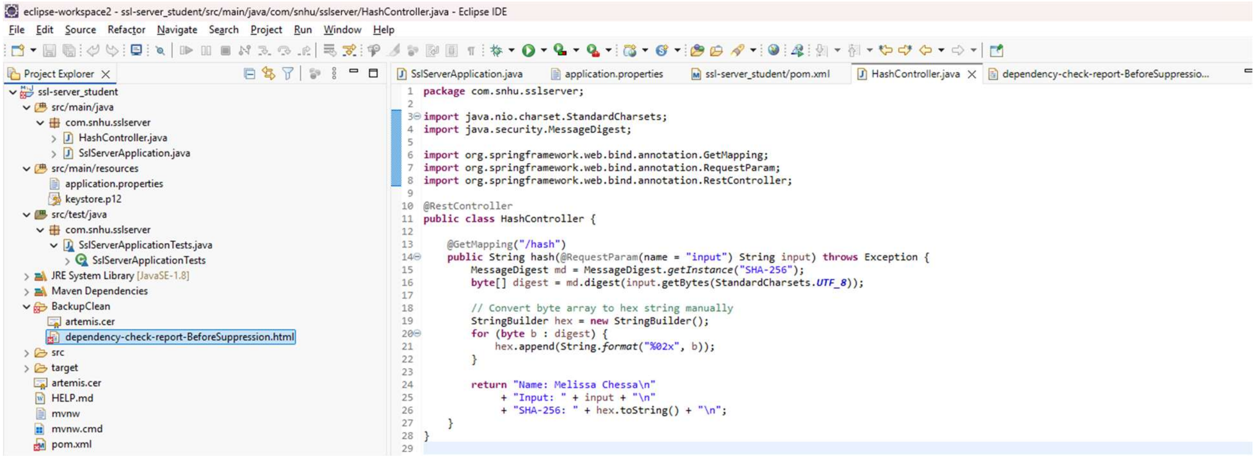
Project: ssl-server

com.snhu:ssl-server:0.0.1-SNAPSHOT

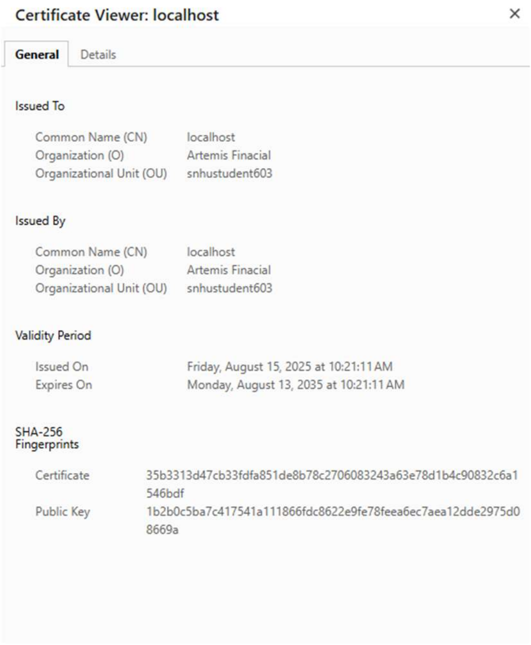
Scan Information (show all):

- *dependency-check version*: 5.3.0
- *Report Generated On*: Sun, 17 Aug 2025 18:38:39 -0400
- *Dependencies Scanned*: 49 (36 unique)
- *Vulnerable Dependencies*: 19
- *Vulnerabilities Found*: 206
- *Vulnerabilities Suppressed*: 9
- ...

6. Functional Testing


eclipse-workspace2 - ssl-server_student/src/main/java/com/snhu/sslserver/HashController.java - Eclipse IDE
File
Edit
Source
Refactor
Navigate
Search
Project
Run
Window
Help
Project Explorer
ssl-server_student
src/main/java
com.snhu.sslserver
HashController.java
SslServerApplication.java
src/main/resources
application.properties
keystore.p12
src/test/java
com.snhu.sslserver
SslServerApplicationTests.java
SslServerApplicationTests
JRE System Library [JavaSE-1.8]
Maven Dependencies
BackupClean
artemis.cer
dependency-check-report-BeforeSuppression.html
src
target
artemis.cer
HELP.md
mvnw
mvnw.cmd
pom.xml
SslServerApplication.java
application.properties
ssl-server_student/pom.xml
HashController.java
dependency-check-report-BeforeSuppressio...

```java
package com.snhu.sslserver;

import java.nio.charset.StandardCharsets;
import java.security.MessageDigest;

import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.annotation.RestController;

@RestController
public class HashController {

    @GetMapping("/hash")
    public String hash(@RequestParam(name = "input") String input) throws Exception {
        MessageDigest md = MessageDigest.getInstance("SHA-256");
        byte[] digest = md.digest(input.getBytes(StandardCharsets.UTF_8));

        // Convert byte array to hex string manually
        StringBuilder hex = new StringBuilder();
        for (byte b : digest) {
            hex.append(String.format("%02x", b));
        }

        return "Name: Melissa Chessa\n"
             + "Input: " + input + "\n"
             + "SHA-256: " + hex.toString() + "\n";
    }
}
```



Certificate Viewer: localhost

Issued To

Common Name (CN)	localhost
Organization (O)	Artemis Finacial
Organizational Unit (OU)	snhustudent603

Issued By

Common Name (CN)	localhost
Organization (O)	Artemis Finacial
Organizational Unit (OU)	snhustudent603

Validity Period

Issued On	Friday, August 15, 2025 at 10:21:11 AM
Expires On	Monday, August 13, 2035 at 10:21:11 AM

SHA-256 Fingerprints

Certificate	35b3313d47cb33fdfa851de8b78c2706083243a63e78d1b4c90832c6a1546bdf
Public Key	1b2b0c5ba7c417541a111866fdc8622e9fe78feea6ec7aea12dde2975d08669a

7. Summary

[In this project I refactored Artemis Financial’s starter application to add a checksum verification endpoint and secure it over HTTPS. I created a self-signed certificate and configured Spring Boot to serve the app on port 8443 using TLS. I implemented a SHA-256 hashing route that returns my name, the input string, and the computed checksum, and I verified functionality in the browser with multiple distinct inputs. I then ran OWASP Dependency-Check to confirm my refactor did not introduce additional vulnerabilities; I documented results and suppressed only non-applicable findings (test-only, API-only, or container-provided). Finally, I reviewed the code for logic, syntax, and basic security issues and validated that the server starts cleanly and the endpoint behaves as expected over HTTPS. This end-to-end process demonstrates refactoring to meet security requirements and compliance with the testing steps called for in the assignment.]

8. Industry Standard Best Practices

[

Encrypt in transit (HTTPS/TLS): I enforced HTTPS with a keystore so all client-server traffic is encrypted. In production, certificates would be CA-signed and renewed automatically.

Use modern integrity primitives: I used SHA-256 for checksums and avoided deprecated algorithms (e.g., MD5, SHA-1).

Shift-left security testing: I integrated OWASP Dependency-Check into the Maven build to surface known third-party risks after code changes. Findings were triaged; only false/non-applicable items were suppressed, while real framework vulnerabilities remain documented for a future upgrade cycle.

Least surprise & separation of concerns: The hashing logic lives in a dedicated controller and the app bootstrap remains minimal. This improves readability, testing, and maintenance.

Secure defaults & input handling: The endpoint expects explicit input and returns deterministic output; optional defaults can be provided safely for demonstration without processing untrusted expressions.

Documentation & evidence: I captured screenshots for certificate creation, HTTPS access, checksum results, build logs, and the scan report to provide an auditable trail of changes and tests..]