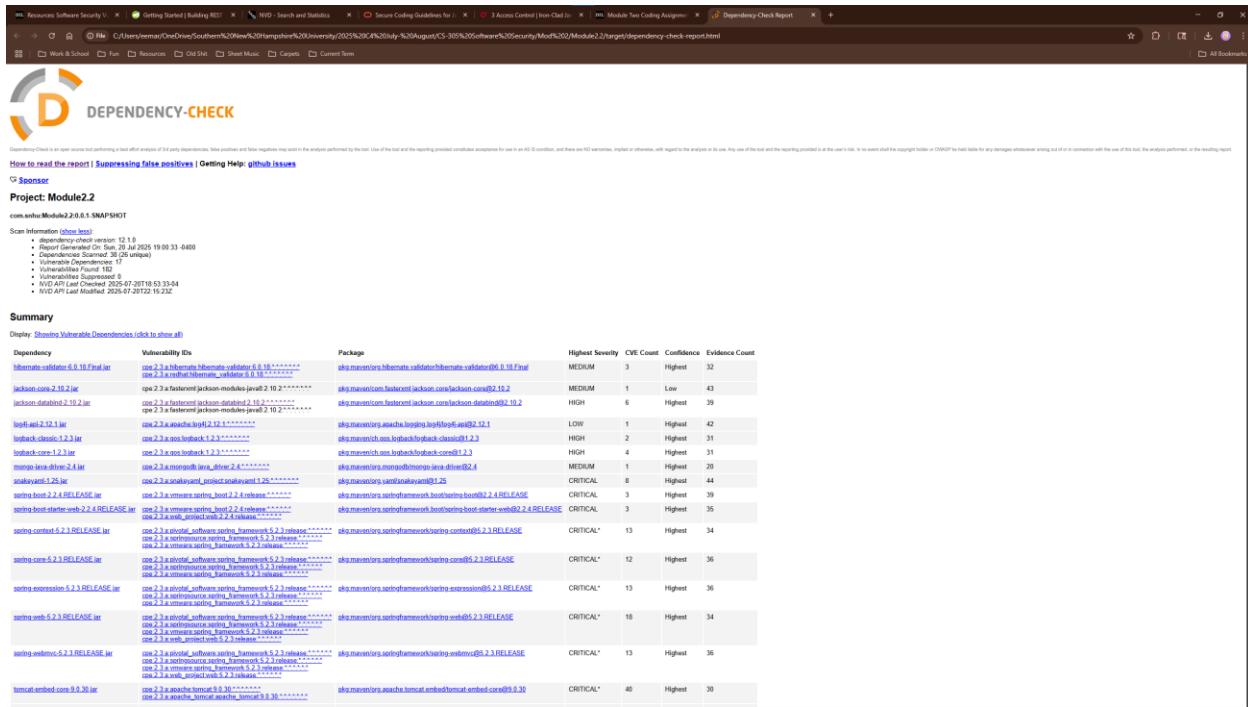


Elizabeth Marticello

CS 305 Module Two Coding Assignment Template

1. Run 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 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 Contributors 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](#)

Scanned Project: **Module2.2**

Scan Information ([Show less](#))

Dependency	Vulnerability IDs	Package	Highest Severity	CVE Count	Confidence	Evidence Count
hibernate.validator-6.0.10.Final.jar	one.2.3.x.hibernate.validator:hibernate-validator@6.0.10.Final one.2.3.x.refactor.hibernate_validator@6.0.10.Final	org.hibernate.validator:hibernate-validator@6.0.10.Final	MEDIUM	3	Highest	32
jackson.core-2.10.2.jar	com.fasterxml.jackson-modules:jackson-module-javatime@2.10.2+***	com.fasterxml.jackson.core:jackson-core@2.10.2	MEDIUM	1	Low	43
jackson-databind-2.10.2.jar	com.fasterxml.jackson.modules:jackson-module-javatime@2.10.2+***	com.fasterxml.jackson.core:jackson-databind@2.10.2	HIGH	6	Highest	39
log4j-api-2.13.1.jar	org.apache.logging.log4j:log4j-api@2.13.1+***	org.apache.logging.log4j:log4j-api@2.13.1	LOW	1	Highest	42
logback-classic-1.2.3.jar	ch.qos.logback:logback-classic@1.2.3+***	ch.qos.logback:logback-classic@1.2.3	HIGH	2	Highest	31
logback-core-1.2.3.jar	ch.qos.logback:logback-core@1.2.3+***	ch.qos.logback:logback-core@1.2.3	HIGH	4	Highest	31
monzo-java-client-2.4.1.jar	com.monzo.java:monzo-client@2.4.1+***	com.monzo.java:monzo-client@2.4.1	MEDIUM	1	Highest	20
stakeholder-2.25.jar	com.zenika.stakeholder:stakeholder@2.25+***	com.zenika.stakeholder:stakeholder@2.25	Critical	8	Highest	44
spring-boot-2.2.4.RELEASE.jar	org.springframework.boot:spring-boot@2.2.4.RELEASE	org.springframework:spring-framework-boot@2.2.4.RELEASE	Critical	3	Highest	39
spring-boot-starter-web-2.2.4.RELEASE.jar	org.springframework.boot:spring-boot-starter-web@2.2.4.RELEASE org.springframework.boot:spring-boot-starter@2.2.4.RELEASE org.springframework.boot:type-safe-project@2.2.4.RELEASE	org.springframework:spring-framework-boot@2.2.4.RELEASE	Critical	3	Highest	35
spring-context-5.2.1.RELEASE.jar	org.springframework:spring-context@5.2.1.RELEASE org.springframework:spring-aop@5.2.1.RELEASE org.springframework:spring-beans@5.2.1.RELEASE	org.springframework:spring-framework-spring-context@5.2.1.RELEASE	Critical	13	Highest	34
spring-core-5.2.3.RELEASE.jar	org.springframework:spring-core@5.2.3.RELEASE org.springframework:spring-aop@5.2.3.RELEASE org.springframework:spring-beans@5.2.3.RELEASE	org.springframework:spring-framework-spring-core@5.2.3.RELEASE	Critical	12	Highest	36
spring-expression-5.2.3.RELEASE.jar	org.springframework:spring-expression@5.2.3.RELEASE org.springframework:spring-core@5.2.3.RELEASE org.springframework:spring-beans@5.2.3.RELEASE	org.springframework:spring-framework-spring-expression@5.2.3.RELEASE	Critical	13	Highest	36
spring-math-5.2.3.RELEASE.jar	org.springframework:spring-math@5.2.3.RELEASE org.springframework:spring-core@5.2.3.RELEASE org.springframework:spring-beans@5.2.3.RELEASE	org.springframework:spring-framework-spring-math@5.2.3.RELEASE	Critical	18	Highest	34
spring-amqp-5.2.3.RELEASE.jar	org.springframework:spring-amqp@5.2.3.RELEASE org.springframework:spring-core@5.2.3.RELEASE org.springframework:spring-beans@5.2.3.RELEASE	org.springframework:spring-framework-spring-amqp@5.2.3.RELEASE	Critical	13	Highest	36
tomcat-embed-compat-9.0.30.jar	org.apache.tomcat:tomcat-embed-compat@9.0.30 org.apache.tomcat:tomcat-embed@9.0.30	org.apache.tomcat:tomcat-embed-compat@9.0.30	Critical	40	Highest	30

Project: Module2.2

com.snhu:Module2.2:0.0.1-SNAPSHOT

Scan Information ([Show less](#)):

- *dependency-check version: 12.1.0*
- *Report Generated On: Sun, 20 Jul 2025 19:00:33 -0400*
- *Dependencies Scanned: 38 (26 unique)*
- *Vulnerable Dependencies: 17*
- *Vulnerabilities Found: 182*
- *Vulnerabilities Suppressed: 0*
- *NVD API Last Checked: 2025-07-20T18:53:33-04*
- *NVD API Last Modified: 2025-07-20T22:15:23Z*

2. Document Results

There were 38 dependencies scanned, 26 of which were unique. Within these 26 unique dependencies there were 17 vulnerable dependencies and an overall 182 vulnerabilities found.

The 26 unique dependencies are listed below with their codes and descriptions:

- classmate-1.5.1.jar
 - o Code: null. No known vulnerabilities.
 - o Library for introspecting types with full generic information including resolving of field and method types.
- hibernate-validator-6.0.18.Final.jar
 - o Code: [CVE-2023-1932](#)
 - o A flaw was found in hibernate-validator's 'isValid' method in the org.hibernate.validator.internal.constraintvalidators.hv.SafeHtmlValidator class, which can be bypassed by omitting the tag ending in a less-than character. Browsers may render an invalid html, allowing HTML injection or Cross-Site-Scripting (XSS) attacks.
- jackson-annotations-2.10.2.jar
- jackson-core-2.10.2.jar
 - o Code: [CVE-2025-49128](#)
 - o Core Jackson processing abstractions (aka Streaming API), implementation for JSON
- jackson-databind-2.10.2.jar
 - o Code: CVE-2023-35116, CVE-2021-46877, CVE-2022-42004, CVE-2022-42003, CVE-2020-36518, CVE-2020-25649
 - o General data-binding functionality for Jackson: works on core streaming API
- jakarta.annotation-api-1.3.5.jar
 - o Core annotations used for value types, used by Jackson data binding package.
- jakarta.validation-api-2.0.2.jar
 - o Jakarta Bean Validation API
- jboss-logging-3.4.1.Final.jar
 - o The JBoss Logging Framework
- jul-to-slf4j-1.7.30.jar
 - o JUL to SLF4J bridge
- log4j-api-2.12.1.jar
 - o Code: CVE-2021-44832, CVE-2021-45105, CVE-2021-45046, CVE-2021-44228, CVE-2020-9488
 - o The Apache Log4j API
- log4j-to-slf4j-2.12.1.jar
 - o The Apache Log4j binding between Log4j 2 API and SLF4J.
- logback-classic-1.2.3.jar
 - o Code: [CVE-2023-6378](#), [CVE-2021-42550](#)
 - o A serialization vulnerability in logback receiver component part of logback version 1.4.11 allows an attacker to mount a Denial-Of-Service attack by sending poisoned data.

- In logback version 1.2.7 and prior versions, an attacker with the required privileges to edit configurations files could craft a malicious configuration allowing to execute arbitrary code loaded from LDAP servers.
- logback-core-1.2.3.jar
 - Code: [CVE-2023-6378](#), [CVE-2021-42550](#)
 - A serialization vulnerability in logback receiver component part of logback version 1.4.11 allows an attacker to mount a Denial-Of-Service attack by sending poisoned data.
 - In logback version 1.2.7 and prior versions, an attacker with the required privileges to edit configurations files could craft a malicious configuration allowing to execute arbitrary code loaded from LDAP servers.
- mongo-java-driver-2.4.jar
 - Java Driver for MongoDB
- slf4j-api-1.7.30.jar
 - The slf4j API
- snakeyaml-1.25.jar
 - Code: CVE-2022-1471, CVE-2022-41854, CVE-2022-38752, CVE-2022-38751, CVE-2022-38750, CVE-2022-38749, CVE-2022-25857, CVE-2017-18640
 - YAML 1.1 parser and emitter for Java
- spring-boot-2.2.4.RELEASE.jar
 - Code: [CVE-2023-20883](#), [CVE-2023-20873](#), [CVE-2022-27772](#)
 - Spring Boot
- spring-boot-starter-web-2.2.4.RELEASE.jar
 - Code: [CVE-2023-20883](#), [CVE-2023-20873](#), [CVE-2022-27772](#)
 - Starter for building web, including RESTful, applications using Spring MVC. Uses Tomcat as the default embedded container
- spring-context-5.2.3.RELEASE.jar
 - Code: CVE-2024-22259, CVE-2023-20863, CVE-2023-20861, CVE-2022-22971, CVE-2022-22970, CVE-2022-22968, CVE-2022-22965, CVE-2022-22950, CVE-2021-22060, CVE-2021-22096, CVE-2021-22118, CVE-2020-5421, CVE-2016-1000027
 - Spring Context
- spring-core-5.2.3.RELEASE.jar
 - Code: CVE-2024-22259, CVE-2023-20863, CVE-2023-20861, CVE-2022-22971, CVE-2022-22970, CVE-2022-22968, CVE-2022-22965, CVE-2022-22950, CVE-2021-22060, CVE-2021-22096, CVE-2021-22118, CVE-2020-5421, CVE-2016-1000027
 - Spring Core
- spring-expression-5.2.3.RELEASE.jar
 - Code: CVE-2024-22259, CVE-2023-20863, CVE-2023-20861, CVE-2022-22971, CVE-2022-22970, CVE-2022-22968, CVE-2022-22965, CVE-2022-22950, CVE-2021-22060, CVE-2021-22096, CVE-2021-22118, CVE-2020-5421, CVE-2016-1000027
 - Spring Expression Language (SpEL)
- spring-web-5.2.3.RELEASE.jar

- Code: CVE-2024-22259, CVE-2023-20863, CVE-2023-20861, CVE-2022-22971, CVE-2022-22970, CVE-2022-22968, CVE-2022-22965, CVE-2022-22950, CVE-2021-22060, CVE-2021-22096, CVE-2021-22118, CVE-2020-5421, CVE-2016-1000027
 - Spring Web
- spring-webmvc-5.2.3.RELEASE.jar
 - Code: CVE-2024-22259, CVE-2023-20863, CVE-2023-20861, CVE-2022-22971, CVE-2022-22970, CVE-2022-22968, CVE-2022-22965, CVE-2022-22950, CVE-2021-22060, CVE-2021-22096, CVE-2021-22118, CVE-2020-5421, CVE-2016-1000027
 - Spring Web MVC
- tomcat-embed-core-9.0.30.jar
 - Code: [CVE-2025-49125](#)
 - Core Tomcat implementation
- tomcat-embed-el-9.0.30.jar
 - Core Tomcat implementation
- tomcat-embed-websocket-9.0.30.jar
 - Code: [CVE-2025-49125](#)
 - Core Tomcat implementation

3. Analyze Results

- Identify the best solutions for addressing dependencies in the code base.

Part of being a great software engineer or developer is to be efficient. Even with my limited knowledge of this project I can tell that there's a lot of unnecessary stuff going on. My biggest recommendation would be to cut down on the unnecessary portions of this project to make it more efficient and limit the copious amounts of vulnerabilities. If cutting down the extra parts of this project is unavailable my next recommendation would be to upgrade to newer versions of the dependencies. A good example of this would be the Jackson-core. There seems to be an upgrade available that should benefit the project.

- Why should you filter false positives from the dependency-check tool?

Filtering false positives has many benefits. The main reason would be to limit the number of resources spent searching for solutions to a non-existent problem. Application projects are often struggling to stay within time restraints and chasing fake problems won't help. A time-smart and efficient team would know how to prioritize issues, knowing how to categorize non-issues will help the team succeed on time.