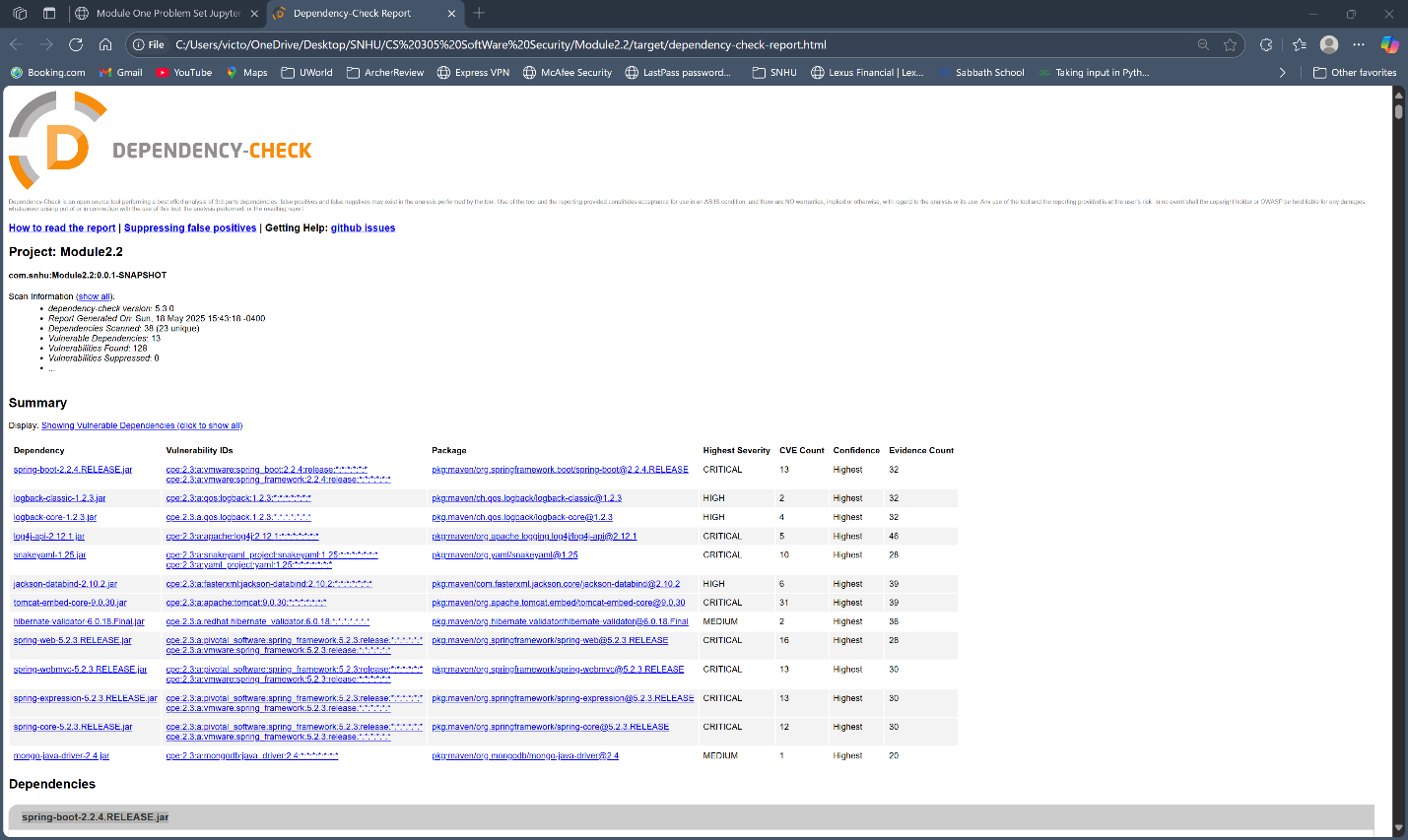
# Victor Ngetich

**CS 305**

**5/18/2025**

# CS 305 Module Two Coding Assignment Template

## Run Dependency Check



## Document Results

* Spring-boot-2.2.4.RELEASE.jar
* Description; Spring Boot
* License: Apache License, Version 2.0: <https://www.apache.org/licenses/LICENSE-2.0>
* Logback-classic-1.2.3.jar
* Description: logback-classic module
* License:

<http://www.eclipse.org/legal/epl-v10.html,%20http://www.gnu.org/licenses/old-licenses/lgpl-2.1.html>

* Logback-core-1.2.3.jar
* Description: logback-core module
* License:

<http://www.eclipse.org/legal/epl-v10.html,%20http://www.gnu.org/licenses/old-licenses/lgpl-2.1.html>

* Log4j-api-2.12.1 jar
* Description: The Apache Log4j API
* License: <https://www.apache.org/licenses/LICENSE-2.0.txt>
* Snakeyaml-1.25.jar
* Description: YAML 1.1 parser and emitter for java
* License : Apache License, Version 2.0: <http://www.apache.org/licenses/LICENSE-2.0.txt>
* Jackson- databind-2.10.2.jar
* Description: General data-binding functionality for Jackson: works on core streaming API
* License: <http://www.apache.org/licenses/LICENSE-2.0.txt>
* tomcat-embed-core-9.0.30.jar
* Description; Core tomcat implementation
* License: Apache License, Version 2.0: <http://www.apache.org/licenses/LICENSE-2.0.txt>
* Hibernate-validator-6.0.18.Final.jar
* Description: Hibernate’s Bean Validation (JSR-380) reference implementation
* License: <http://www.apache.org/licenses/LICENSE-2.0.txt>
* Spring-web-5.2.3.RELEASE.jar
* Description Spring Web
* License: Apache License, Version 2.0: <https://www.apache.org/licenses/LICENSE-2.0>
* Spring-webmvc-5.2.3.RELEASE.jar
* Description: Spring Web MVC
* License: Apache License, Version 2.0: <https://www.apache.org/licenses/LICENSE-2.0>
* Spring-expression-5.2.3.RELEASE.jar
* Description: Spring Expression Language (SpEL)
* License: Apache License, Version 2.0: <https://www.apache.org/licenses/LICENSE-2.0>
* Spring-core-5.2.3.RELEASE.jar
* Description: Spring Core
* License: Apache License, Version 2.0: <https://www.apache.org/licenses/LICENSE-2.0>
* Mongo-java-driver-2.4.jar
* Description: Java Driver for MongoDB
* License: The Apache Software License, Version 2.0: <http://www.apache.org/licenses/LICENSE-2.0.txt>

## Analyze Results

After analyzing the results from the dependency-check report, it is evident that the current codebase contains multiple outdated and vulnerable libraries. The report flagged a total of thirteen dependencies with known security issues, including components from the Spring Framework (e.g., spring-webmvc, spring-core, and spring-expression), as well as libraries like logback, jackson-databind, and the MongoDB Java driver. Several of these components are associated with publicly disclosed vulnerabilities documented in the Common Vulnerabilities and Exposures (CVE) system, including high-risk entries such as CVE-2016-1000027 and CVE-2021-42550. According to the National Vulnerability Database (NVD), these vulnerabilities could potentially lead to remote code execution, improper input validation, or denial of service if exploited.

To address these issues, the best solution is to upgrade all flagged dependencies to their latest stable and secure versions. For example, the Spring Framework should be updated from version 5.2.3.RELEASE to a more recent version that no longer carries the vulnerabilities listed in the report. In addition to version upgrades, it is essential to adopt a proactive security strategy by integrating automated tools like OWASP Dependency-Check, Dependabot, or Snyk into the CI/CD pipeline. These tools help ensure that new vulnerabilities are identified and addressed promptly during development.

It is also important to filter out false positives reported by dependency scanning tools. Not every flagged vulnerability is necessarily exploitable in the context of a specific project. Some may only apply if certain features or methods are used, which may not be the case in the actual implementation. Filtering out these irrelevant alerts helps development teams focus on true risks, reduces unnecessary code changes, and improves the efficiency of the remediation process.

In summary, maintaining secure dependencies requires continuous monitoring, timely upgrades, and careful evaluation of vulnerability reports. By following industry-standard guidelines provided through CVE entries and NVD scoring, developers can prioritize remediation efforts effectively and strengthen the overall security posture of their applications.

**REFERENCE**

Mirhosseini, S., & Parnin, C. (2017, October). Can automated pull requests encourage software developers to upgrade out-of-date dependencies?. In 2017 32nd IEEE/ACM international conference on automated software engineering (ASE) (pp. 84-94).

Herzog, P. (2003). Open-source security testing methodology manual. Institute for Security and Open Methodologies (ISECOM).