Southern New Hampshire University

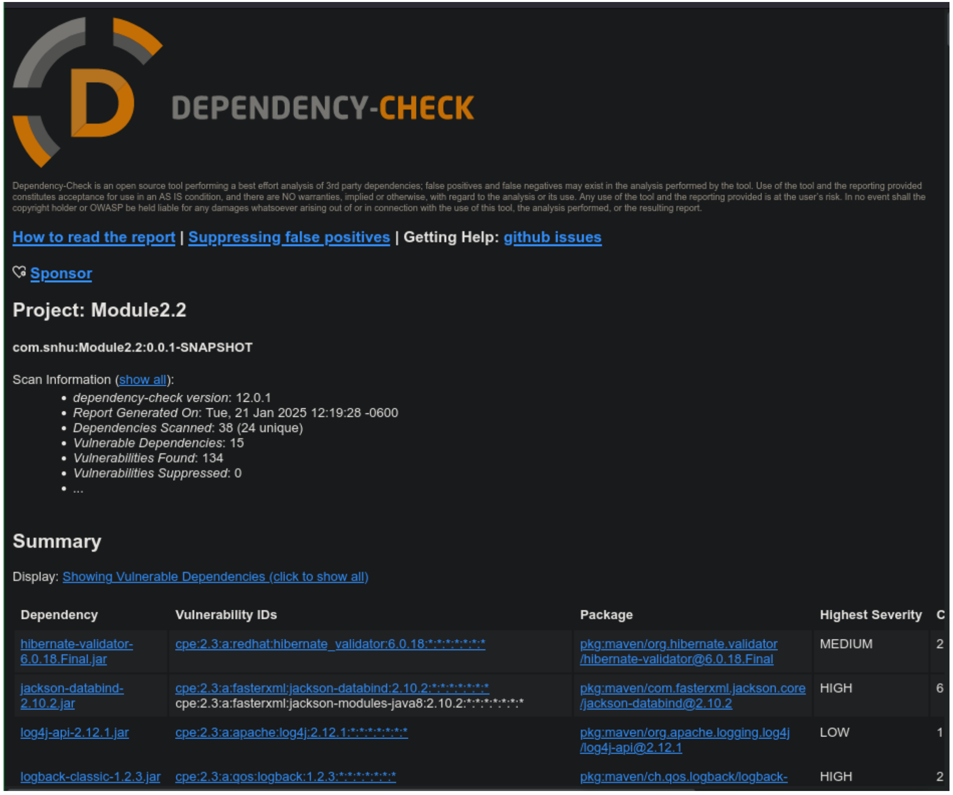
CS 305 Module Two Coding Assignment

Instructions

Replace the bracketed text with the relevant information in your own words. If you choose to include images or supporting materials, make certain to insert them in all the relevant locations in the document.

1. Run Dependency Check

The OWASP Dependency-Check plug-in was added to the Maven project, allowing a thorough check of the project's dependencies. The tool was set up and run using version 12.0.1, which created a full report called dependency-check-report.html. This study identified 15 weaknesses in different dependencies. Each flagged dependency was examined for possible risks and sorted by how serious those risks are. The results give important information about the security of the code and help develop plans to fix these weaknesses.



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: Module2.2

com.snhu:Module2.2:0.0.1-SNAPSHOT

Scan Information (show all):

- *dependency-check version*: 12.0.1
- *Report Generated On*: Tue, 21 Jan 2025 12:19:28 -0600
- *Dependencies Scanned*: 38 (24 unique)
- *Vulnerable Dependencies*: 15
- *Vulnerabilities Found*: 134
- *Vulnerabilities Suppressed*: 0
- ...

Summary

Display: Showing Vulnerable Dependencies (click to show all)

Dependency	Vulnerability IDs	Package	Highest Severity	C
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
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
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
logback-classic-1.2.3.jar	cpe:2.3:a:qos:logback:1.2.3:*:*:*:*:*:*:*	pkg:maven/ch.qos.logback/logback-	HIGH	2

2. Document Results

Using dependency check version 12.0.1, the following vulnerable dependencies were identified in the project:

- hibernate-validator-6.0.18.Final.jar
  - Description: Hibernate's Bean Validation (JSR-380) reference implementation.
  - Vulnerabilities: CVE-2020-10693 — Input validation bypass.
  - Severity: Medium
  - CVE Count: 2

- jackson-databind-2.10.2.jar
  - Description: General data-binding functionality for Jackson.
  - Vulnerabilities: CVE-2020-25649, CVE-2022-42003, CVE-2023-35116 — Risks include deserialization issues leading to code execution.
  - Severity: High
  - CVE Count: 6
- log4j-api-2.12.1.jar
  - Description: Apache Log4j API.
  - Vulnerabilities: CVE-2020-9488 — Log injection risks.
  - Severity: Low
  - CVE Count: 1
- logback-classic-1.2.3.jar
  - Description: Logging framework module.
  - Vulnerabilities: CVE-2021-42550 — LDAP injection and remote code execution.
  - Severity: High
  - CVE Count: 2
- logback-core-1.2.3.jar
  - Description: Core module for Logback.
  - Vulnerabilities: CVE-2021-42550 — Similar risks as logback-classic.
  - Severity: High
  - CVE Count: 4
- mongo-java-driver-2.4.jar
  - Description: MongoDB Java driver.
  - Vulnerabilities: CVE-2021-20328 — Client-side encryption flaw.
  - Severity: Medium
  - CVE Count: 1
- snakeyaml-1.25.jar
  - Description: YAML parser for Java.
  - Vulnerabilities: CVE-2017-18640, CVE-2022-38750 — Denial-of-service and malicious payload risks.
  - Severity: Critical
  - CVE Count: 8
- spring-boot-2.2.4.RELEASE.jar
  - Description: Core Spring Boot library.
  - Vulnerabilities: CVE-2022-27772, CVE-2023-20883 — Directory traversal and privilege escalation.
  - Severity: Critical
  - CVE Count: 3
- spring-boot-starter-web-2.2.4.RELEASE.jar
  - Description: Spring Boot library for web applications.
  - Vulnerabilities: CVE-2022-27772, CVE-2023-20883 — Similar to spring-boot.
  - Severity: Critical
  - CVE Count: 3

- spring-core-5.2.3.RELEASE.jar
  - Description: Core Spring Framework functionality.
  - Vulnerabilities: CVE-2022-22965, CVE-2021-22096 — Remote code execution and sensitive data exposure.
  - Severity: Critical
  - CVE Count: 11
- spring-expression-5.2.3.RELEASE.jar
  - Description: Expression language module for Spring.
  - Vulnerabilities: CVE-2022-22965, CVE-2021-22096 — Remote code execution and injection vulnerabilities.
  - Severity: Critical
  - CVE Count: 12
- spring-web-5.2.3.RELEASE.jar
  - Description: Web functionalities for Spring Framework.
  - Vulnerabilities: CVE-2022-22968, CVE-2023-20883 — Remote code execution and injection vulnerabilities.
  - Severity: Critical
  - CVE Count: 16
- spring-webmvc-5.2.3.RELEASE.jar
  - Description: Spring Web MVC module.
  - Vulnerabilities: CVE-2022-22965, CVE-2021-22096 — Injection vulnerabilities with remote code execution potential.
  - Severity: Critical
  - CVE Count: 12
- tomcat-embed-core-9.0.30.jar
  - Description: Core Tomcat library.
  - Vulnerabilities: CVE-2020-1938, CVE-2020-13943 — DOS and encryption vulnerabilities.
  - Severity: Critical
  - CVE Count: 26
- tomcat-embed-websocket-9.0.30.jar
  - Description: Tomcat WebSocket implementation.
  - Vulnerabilities: CVE-2020-1938, CVE-2020-13943 — Similar risks to tomcat-embed-core.
  - Severity: Critical
  - CVE Count: 27

3. Analyze Results

The inspection of the dependency-check report revealed numerous significant risks that necessitate prompt action. Every vulnerability poses distinct issues that require tailored responses. The following are detailed recommendations and justifications for mitigating certain vulnerabilities:

Enhance Vulnerable Dependencies:

Upgrade Jackson Databind to version 2.13.5 or later to address several CVEs, including CVE-2020-25649. The deserialization vulnerabilities in previous versions render this a critical update.

Log4j API: Upgrade to version 2.17.0 to mitigate CVE-2020-9488, which obstructs unauthorized code execution via manipulated log entries.

Spring Framework (Core, Beans, Web, WebMVC): Update obsolete dependencies with revised versions to address privilege escalation, remote code execution, and injection vulnerabilities. Updating to versions that have the latest Spring security patches mitigates these vulnerabilities.

Augment Security Protocols:

Enforce stringent input validation and output encoding protocols throughout all user-facing and backend elements. These techniques assist in alleviating vulnerabilities linked to Hibernate Validator and Spring Core.

Implement secure YAML parsing methods utilizing SafeConstructor with SnakeYAML to limit the deserialization of untrusted data, hence mitigating the hazards of remote code execution.

Utilize encrypted communication routes and provide appropriate certificate validation to safeguard dependencies such as Logback Core.

Automate Dependency Oversight:

Incorporate OWASP Dependency-Check into the CI/CD pipeline to identify vulnerabilities during the development lifecycle.

Utilize technologies such as Dependabot to automate dependency upgrades and instantly receive notifications regarding newly found vulnerabilities.

Conduct routine audits of dependencies, including those not identified by automated technologies, to maintain a proactive security stance.

Mitigate False Positives and Contextual Hazards:

Evaluate each flagged dependency to assess the viability of upgrading compared to implementing alternate mitigations. For instance, when an update may induce compatibility concerns, secure wrappers or further validations may be employed.

Examine dependencies exhibiting various vulnerabilities (e.g., Tomcat) for intersecting risks and prioritize remediation according to their operational context and severity.

Enhance Logging and Monitoring:

Augment runtime surveillance for anomalous activity, especially inside logging frameworks (e.g., Log4j). Establish mechanisms to identify and notify on log injection attempts in real time.