# CS 305 Module Two Coding Assignment Template

## 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.

## Run Dependency Check

A screenshot of a computer screen

AI-generated content may be incorrect.

## Document Results

Based on the dependency check, I found the following vulnerabilities.

* jackson-databind-2.15.4.jar (Medium, CVE Count: 1, Evidence Count: 41)
  + CVE-2023-35116
    - This allows attackers to cause a denial of service or other unspecified impact via a crafted object that uses cyclic dependencies.
* json-smart-2.5.1.jar (High, CVE Count: 1, Evidence Count: 51)
  + CVE-2024-57699(OSSINDEX)
    - A security issue found in Netplex Json-smart 2.5.0 through 2.5.1. When loading a specially crafted JSON input, containing a large number of “{“, a stack exhaustion can be triggered which could allow an attacker to cause a Denial of Service (DoS).
* logback-core-1.4.14.jar (Medium, CVE Count: 2, Evidence Count: 36)
  + CVE-2024-12798(OSSINDEX)
    - In Java applications allows attacker to execute arbitrary code by compromising an existing logback configuration file or by injecting an environment variable before program execution.
    - Malicious logback configuration files can allow the attacker to execute arbitrary code using the JaninoEventEvaluator extension.
    - A successful attack requires the user to have write access to a configuration file. Alternatively the attacker could inject a malicious environment variable pointing to a malicious configuration file.
  + CVE-2024-12801 (OSSINDEX)
    - Allows an attacker to forge requests by compromising logback configuration files in XML
    - The attack involves the modification of DOCTYPE declaration in XML configuration files.
* spring-context-6.1.6.jar(Medium, CVE Count: 2, Evidence Count: 35)
  + CVE-2024-38820
    - Locale dependent exceptions that could potentially result in fields not protected as expected.
  + CVE-2025-22233(OSSINDEX)
    - Cases where it is possible to bypass the disallowedField checks.
* spring-core-6.1.6.jar (High, CVE Count: 3, Evidence Count: 41)
  + CVE-2025-41249(OSSINDEX)
    - The Spring Framework annotation detection mechanism may not correctly resolve annotations on methods within type hierarchies with a parameterized super type with unbounded generics. This can be an issue if such annotations are used for authorization decisions.
    - Your application may be affected by this if you are using Spring Security’s @EnableMethodSecurity feature.
  + CVE-2025-41242(OSSINDEX)
    - Spring Framework MVC applications can be vulnerable to a “Path Traversal Vulnerability” when deployed on a non-compliant Servlet container.
    - An application can be vulnerable when all the following are true:
      * The application is deployed as a WAR or with an embedded Servlet container.
      * The Servlet container does not reject suspicious sequences.
      * The application serves static resources with Spring resource handling.
  + CVE-2024-38820
    - The fix for CVE-2022-22968 made disallowedFields patterns in DataBinder case insensitive. However, Spring.toLowerCase() has some Locale dependent exceptions that could potentially result in fields not protected as expected.
* spring-tx-6.1.6.jar (Medium, CVE Count: 1, Evidence Count: 35)
  + CVE-2024-38820
    - The fix for CVE-2022-22968 made disallowedFields patterns in DataBinder case insensitive. However, Spring.toLowerCase() has some Locale dependent exceptions that could potentially result in fields not protected as expected.
* spring-web-6.1.6.jar (High, CVE Count: 3, Evidence Count: 35)
  + CVE-2024-38809(OSSINDEX)
    - Applications that parse ETages from “If-Match” or “If-None-Match” request headers are vulnerable to DoS attack.
  + CVE-2025-41234(OSSINDEX)
    - In Spring Framework versions an application is vulnerable to a reflected file download (RFD) attack when it sets a “Content-Disposition” header with a non-ASCII charset, where the filename attribute is derived from user-supplied input.
    - Specifically, an application is vulnerable when the following is true:
      * The header is prepared with org.springframework.http ContentDisposition.
      * The filename is set via ContentDisposition.Builder#filename(String, Charset).
      * The value for the filename is derived from user-supplied input.
      * The application does not sanitize the user-supplied input.
      * The downloaded content of the response is injected with malicious commands by the attacker.
  + CVE-2024-38820
    - The fix for CVE-2022-22968 made disallowedFields patterns in DataBinder case insensitive. However, Spring.toLowerCase() has some Locale dependent exceptions that could potentially result in fields not protected as expected.
* spring-webmvc-6.1.6.jar (High, CVE Count: 2, Evidence Count: 37)
  + CVE-2024-38816(OSSINDEX)
    - Applications serving static resources through the functional web frameworks WebMvc.jn or WebFlux.fn are vulnerable to path traversal attacks. An attacker can craft malicious HTTP requests and obtain any file on the file system that is also accessible to the process in which the Spring application is running.
    - Specifically an application is vulnerable when both of the following are true:
      * The web application uses RouterFunctions to serve static resources.
      * Resource handling is explicitly configured with a FileSystemResource location.
  + CVE-2024-38820
    - The fix for CVE-2022-22968 made disallowedFields patterns in DataBinder case insensitive. However, Spring.toLowerCase() has some Locale dependent exceptions that could potentially result in fields not protected as expected.
* tomcat-embed-core-10.1.20.jar (Critical, CVE Count: 17, Evidence Count:63)
  + CVE-2024-50379
    - Time-of-check Time-of-use (TOCTOU) Race Condition vulnerability during JSP compilation in Apache Tomcat permits an RCE on case insensitive file systems when the default servlet is enabled for write.
  + CVE-2024-52316
    - Unchecked Error Condition vulnerability in Apache Tomcat. If Tomcat is configured to use a custom Jakarta Authentication ServerAuthContxt component which may throw an exception during the authentication process without explicitly setting an HTTP status to indicate failure.
  + CVE-2024-56337
    - Time-of-check Time-of-use (TOCTOU) Race Condition vulnerability in Apache Tomcat
  + CVE-2025-24813
    - Description: Apache Tomcat contains a path equivalence vulnerability that allows a remote attacker to execute code, disclose information, or inject malicious content via a partial PUT request.
    - If all of the following were true, a malicious user was able to view security sensitive files and/or inject content into those files:
      * - writes enabled for the default servlet (disabled by default)
      * - support for partial PUT (enabled by default)
      * - a target URL for security sensitive uploads that was a sub-directory of a target URL for public uploads
      * - attacker knowledge of the names of security sensitive files being uploaded
      * - the security sensitive files also being uploaded via partial PUT
    - If all of the following were true, a malicious user was able to perform remote code execution:
      * - writes enabled for the default servlet (disabled by default)
      * - support for partial PUT (enabled by default)
      * - application was using Tomcat's file based session persistence with the default storage location
      * - application included a library that may be leveraged in a deserialization attack
  + CVE-2025-31651
    - Improper Neutralization of Escape, Meta, or Control Sequences vulnerability in Apache Tomcat. For a subset of unlikely rewrite rule configurations, it was possible for a specially crafted request to bypass some rewrite rules. If those rewrite rules effectively enforced security constraints, those constraints could be bypassed.
  + CVE-2025-49124
    - Untrusted Search Path vulnerability in Apache Tomcat installer for Windows. During installation, the Tomcat installer for Windows used icacls.exe without specifying a full path.
  + CVE-2024-34750
    - Improper Handling of Exceptional Conditions, Uncontrolled Resource Consumption vulnerability in Apache Tomcat. When processing an HTTP/2 stream, Tomcat did not handle some cases of excessive HTTP headers correctly. This led to a miscounting of active HTTP/2 streams which in turn led to the use of an incorrect infinite timeout which allowed connections to remain open which should have been closed.
  + CVE-2024-38286
    - Allocation of Resources Without Limits or Throttling vulnerability in Apache Tomcat.
  + CVE-2025-31650
    - Improper Input Validation vulnerability in Apache Tomcat. Incorrect error handling for some invalid HTTP priority headers resulted in incomplete clean-up of the failed request which created a memory leak. A large number of such requests could trigger an OutOfMemoryException resulting in a denial of service.
  + CVE-2025-48988
    - Allocation of Resources Without Limits or Throttling vulnerability in Apache Tomcat.
  + CVE-2025-48989
    - Improper Resource Shutdown or Release vulnerability in Apache Tomcat made Tomcat vulnerable to the made you reset attack.
  + CVE-2025-49125
    - Authentication Bypass Using an Alternate Path or Channel vulnerability in Apache Tomcat.  When using PreResources or PostResources mounted other than at the root of the web application, it was possible to access those resources via an unexpected path. That path was likely not to be protected by the same security constraints as the expected path, allowing those security constraints to be bypassed.
  + CVE-2025-52520
    - For some unlikely configurations of multipart upload, an Integer Overflow vulnerability in Apache Tomcat could lead to a DoS via bypassing of size limits.
  + CVE-2025-53506
    - Uncontrolled Resource Consumption vulnerability in Apache Tomcat if an HTTP/2 client did not acknowledge the initial settings frame that reduces the maximum permitted concurrent streams.
  + CVE-2025-46701
    - Improper Handling of Case Sensitivity vulnerability in Apache Tomcat's GCI servlet allows security constraint bypass of security constraints that apply to the pathInfo component of a URI mapped to the CGI servlet.
  + CVE-2025-55668
    - Session Fixation vulnerability in Apache Tomcat via rewrite valve.
  + CVE-2024-54677
    - Uncontrolled Resource Consumption vulnerability in the examples web application provided with Apache Tomcat leads to denial of service.

## Analyze Results

* jackson-databind-2.15.4.jar (Medium, CVE Count: 1, Evidence Count: 41)
  + CVE-2023-35116
    - This vulnerability can be fixed by updating Jackson-databind to 2.16.0 or newer.
* json-smart-2.5.1.jar (High, CVE Count: 1, Evidence Count: 51)
  + CVE-2024-57699(OSSINDEX)
    - This vulnerability can be fixed by updating json-smart to 2.5.2
* logback-core-1.4.14.jar (Medium, CVE Count: 2, Evidence Count: 36)
  + CVE-2024-12798 + CVE-2024-12801
    - These vulnerabilities can be fixed by updating logback-core to 1.5.18
* spring-context-6.1.6.jar(Medium, CVE Count: 2, Evidence Count: 35)
  + For all vulnerability codes
    - These vulnerabilities can be fixed by updating to 6.1.21
* spring-core-6.1.6.jar (High, CVE Count: 3, Evidence Count: 41)
  + For all vulnerability codes
    - These vulnerabilities can be fixed by updating to 6.1.21
* spring-tx-6.1.6.jar (Medium, CVE Count: 1, Evidence Count: 35)
  + For all vulnerability codes
    - These vulnerabilities can be fixed by updating to 6.1.21
* spring-web-6.1.6.jar (High, CVE Count: 3, Evidence Count: 35)
  + For all vulnerability codes
    - These vulnerabilities can be fixed by updating to 6.1.21
* spring-webmvc-6.1.6.jar (High, CVE Count: 2, Evidence Count: 37)
  + For all vulnerability codes
    - These vulnerabilities can be fixed by updating to 6.1.21
* tomcat-embed-core-10.1.20.jar (Critical, CVE Count: 17, Evidence Count:63)
  + For all vulnerability codes
    - These vulnerabilities can be fixed by updating to 11.0.0

1. Also consider why you should filter false positives from the dependency-check tool

Filtering false positives out of the dependency-check can allow the development team to focus on real vulnerabilities. This can help them make the software more secure quicker and can allow them to focus their efforts on high vulnerability threats. Removing false positives also allow the development team to “ignore the noise” since there will be less results to filter through. If the dependency check is given to the stakeholders or another entity it can make it easier to read and removing these false positives can make the report look better.