# CS 305 Module Two Coding Assignment Template

Frederick Brehm

CS 305

Southern New Hampshire University

5/19/2024

## Run Dependency Check

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

## Document Results

There were 13 vulnerable dependencies found:

1.

CVE – 2020 – 10693,

Medium severity

A flaw was found in hibernate version 6.1.2. Final. The flaw allows attackers to bypass input sanitation controls that developers may have put in place. To fix, it seems as if there is an updated version that needs to be installed.

2.

CVE-2020-25649

High severity

A flaw was found in FasterXML Jackson Databind, where it did not have entity expansion secured properly. This flaw allows vulnerability to XML external entity (XXE) attacks. The highest threat from this vulnerability is data integrity.

3.

CVE-2020-9488

Low Severity

Improper validation of certificate with host mismatch in Apache Log4j SMTP appender. This could allow an SMTPS connection to be intercepted by a man-in-the-middle attack which could leak any log messages sent through that appender. Fixed in Apache Log4j 2.12.3 and 2.13.1

4.

CVE-2023-6378

High severity

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.

5.

CVE-2023-6378

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.

6.

[**CVE-2021-200328**](https://nvd.nist.gov/vuln/detail/CVE-2021-20328)

**Medium severity**

**Specific versions of the Java driver that support client-side field level encryption (CSFLE) fail to perform correct host name verification on the KMS server’s certificate. This vulnerability in combination with a privileged networ position active MITM attack could result in interception of traffic between the Java driver and KMS service rendering Filed Level Encryption ineefective.**

7.

CVE-2022-1471

Critical severity

SnakeYaml's Constructor () class does not restrict types which can be instantiated during deserialization. Deserializing yaml content provided by an attacker can lead to remote code execution.

8.

CVE-2023-20873

Critical severity

In Spring Boot versions 3.0.0 - 3.0.5, 2.7.0 - 2.7.10, and older unsupported versions, an application that is deployed to Cloud Foundry could be susceptible to a security bypass.

9.

CVE-2022-22965

Critical severity

A Spring MVC or Spring WebFlux application running on JDK 9+ may be vulnerable to remote code execution (RCE) via data binding. The specific exploit requires the application to run on Tomcat as a WAR deployment. If the application is deployed as a Spring Boot executable jar, i.e., the default, it is not vulnerable to the exploit.

10.

CVE 2016-1000027

Critical severity

Pivotal Spring Framework through 5.3.16 suffers from a potential remote code execution (RCE) issue if used for Java deserialization of untrusted data.

11.

CVE 2022-22965

Critical Severity

A Spring MVC or Spring WebFlux application running on JDK 9+ may be vulnerable to remote code execution (RCE) via data binding. The specific exploit requires the application to run on Tomcat as a WAR deployment.

12.

CVE-2020-1938

Critical severity

When using the Apache JServ Protocol (AJP), care must be taken when trusting incoming connections to Apache Tomcat. Tomcat treats AJP connections as having higher trust than, for example, a similar HTTP connection. If such connections are available to an attacker, they can be exploited in ways that may be surprising.

13.

CVE-2020-1938

Critical severity

When using the Apache JServ Protocol (AJP), care must be taken when trusting incoming connections to Apache Tomcat. Tomcat treats AJP connections as having higher trust than, for example, a similar HTTP connection. If such connections are available to an attacker, they can be exploited in ways that may be surprising.

## Analyze Results

After looking over most of the vulnerable dependencies and reading the description and what they effect, it is easy to see that most of these vulnerabilities have in fact been fixed. Most of the vulnerabilities were caused due to an outdated version of the software. To fix all that needed done was an update to the current version to fix the issues that were being seen. Several vulnerabilities like CVE-2020, CVE-2022, CVE-2016, and CVE-2023, had to do with possible poisoned, untrusted, or exploited data. This typically entails someone illegally obtaining information which can be detrimental to any business or person. These vulnerabilities may have been caught if more static testing was done throughout the development of the software. Unfortunately, though, a development team can test all they want and sometimes mistakes are going to slip through the cracks. However, the severity of what slips through the cracks is why it is so important to continuously test the product to ensure that as many mistakes and defects are caught before going live and jeopardizing the company or worse, people. A false positive is when you expect a certain vulnerability but there is not one at all. This should be filtered out of the vulnerability test to keep the results true to vulnerabilities that need to be fixed. I can see how false positives could help a tester develop their ability in finding these, but false positives do more harm than good. Again, organization and accuracy are crucial in development. The possibility of there being false positives could cause loss of time, resources, and money.

Reference

*Owasp dependency-check*. OWASP Dependency-Check | OWASP Foundation. (n.d.-a). https://owasp.org/www-project-dependency-check/

*You are viewing this page in an unauthorized frame window.* NVD. (n.d.). https://nvd.nist.gov/vuln/search/results?form\_type=Basic&results\_type=overview&query=CVE-2021-20328&search\_type=all&isCpeNameSearch=false