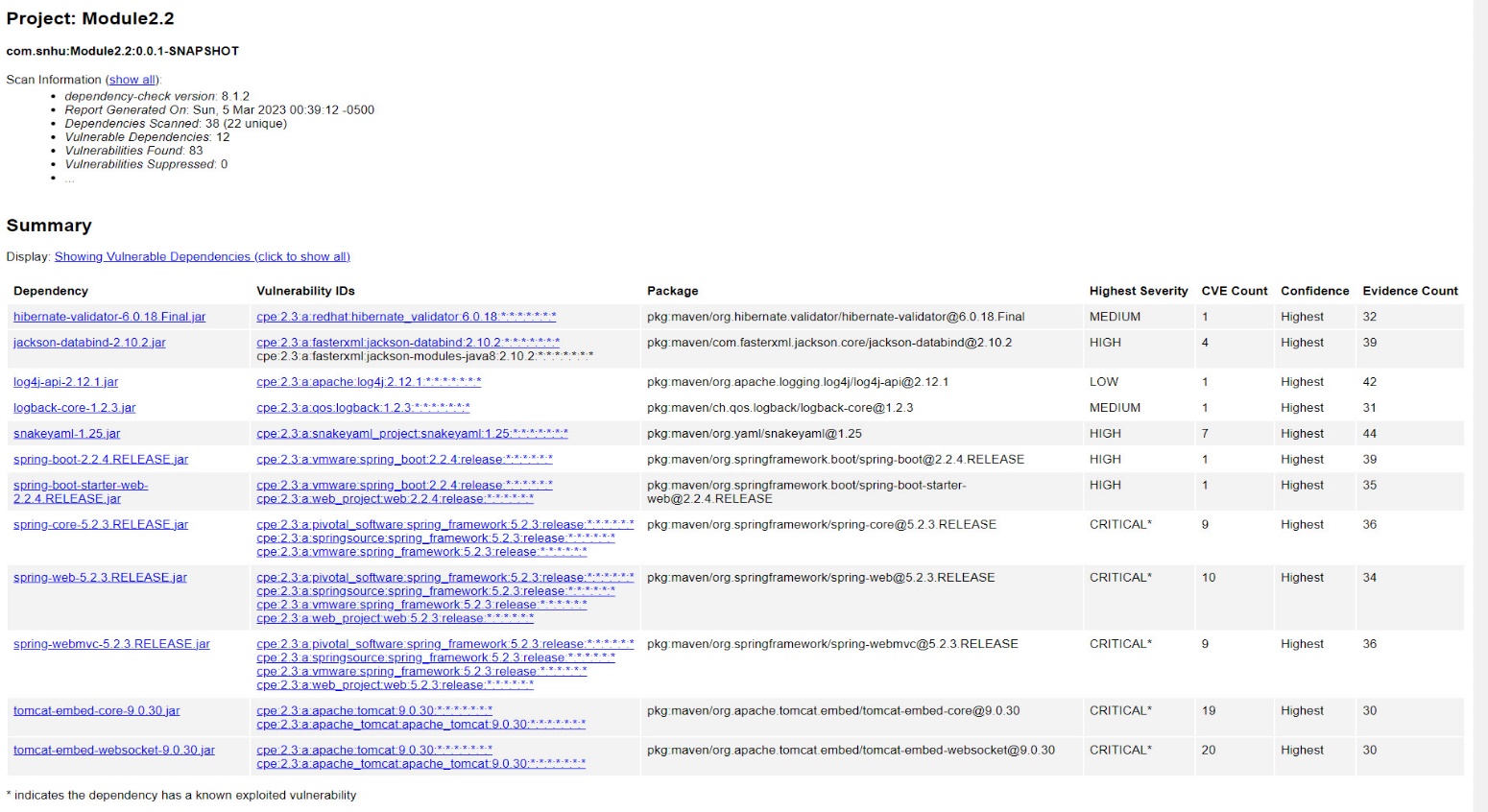
# CS 305 Module Two Coding Assignment

## Run Dependency Check



## Document Results

**Dependency:**

hibernate-validator-6.0.18.Final.jar

**Description:**

Hibernate's Bean Validation (JSR-380) reference implementation.

**Published Vulnerabilities:**

CVE-2020-10693 - A flaw was found in Hibernate Validator version 6.1.2.Final. A bug in the message interpolation processor enables invalid EL expressions to be evaluated as if they were valid. This flaw allows attackers to bypass input sanitation (escaping, stripping) controls that developers may have put in place when handling user-controlled data in error messages.

**Dependency:**

jackson-databind-2.10.2.jar

**Description:**

General data-binding functionality for Jackson: works on core streaming API

**Published Vulnerabilities:**

CVE-2020-25649 - 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.

CVE-2020-36518 - jackson-databind before 2.13.0 allows a Java StackOverflow exception and denial of service via a large depth of nested objects.

CVE-2022-42003 - In FasterXML jackson-databind before 2.14.0-rc1, resource exhaustion can occur because of a lack of a check in primitive value deserializers to avoid deep wrapper array nesting, when the UNWRAP\_SINGLE\_VALUE\_ARRAYS feature is enabled. Additional fix version in 2.13.4.1 and 2.12.17.1

CVE-2022-42004 - In FasterXML jackson-databind before 2.13.4, resource exhaustion can occur because of a lack of a check in BeanDeserializer.\_deserializeFromArray to prevent use of deeply nested arrays. An application is vulnerable only with certain customized choices for deserialization.

**Dependency:**

log4j-api-2.12.1.jar

**Description:**

The Apache Log4j API

**Published Vulnerabilities:**

CVE-2020-9488 - 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

**Dependency:**

logback-core-1.2.3.jar

**Description:**

logback-core module

**Published Vulnerabilities:**

CVE-2021-42550 - 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.

**Dependency:**

snakeyaml-1.25.jar

**Description:**

YAML 1.1 parser and emitter for Java

**Published Vulnerabilities:**

CVE-2017-18640 - The Alias feature in SnakeYAML before 1.26 allows entity expansion during a load operation, a related issue to CVE-2003-1564.

CVE-2022-25857 - The package org.yaml:snakeyaml from 0 and before 1.31 are vulnerable to Denial of Service (DoS) due missing to nested depth limitation for collections.

CVE-2022-38749 - Using snakeYAML to parse untrusted YAML files may be vulnerable to Denial of Service attacks (DOS). If the parser is running on user supplied input, an attacker may supply content that causes the parser to crash by stackoverflow.

CVE-2022-38751 - Using snakeYAML to parse untrusted YAML files may be vulnerable to Denial of Service attacks (DOS). If the parser is running on user supplied input, an attacker may supply content that causes the parser to crash by stackoverflow.

CVE-2022-38752 - Using snakeYAML to parse untrusted YAML files may be vulnerable to Denial of Service attacks (DOS). If the parser is running on user supplied input, an attacker may supply content that causes the parser to crash by stack-overflow.

CVE-2022-41854 - Those using Snakeyaml to parse untrusted YAML files may be vulnerable to Denial of Service attacks (DOS). If the parser is running on user supplied input, an attacker may supply content that causes the parser to crash by stack overflow. This effect may support a denial of service attack.

CVE-2022-38750 - Using snakeYAML to parse untrusted YAML files may be vulnerable to Denial of Service attacks (DOS). If the parser is running on user supplied input, an attacker may supply content that causes the parser to crash by stackoverflow.

**Dependency:**

spring-boot-2.2.4.RELEASE.jar

**Description:**

Spring Boot

**Published Vulnerabilities:**

CVE-2022-27772 - \*\* UNSUPPORTED WHEN ASSIGNED \*\* spring-boot versions prior to version v2.2.11.RELEASE was vulnerable to temporary directory hijacking. This vulnerability impacted the org.springframework.boot.web.server.AbstractConfigurableWebServerFactory.createTempDir method. NOTE: This vulnerability only affects products and/or versions that are no longer supported by the maintainer.

**Dependency:**

spring-boot-starter-web-2.2.4.RELEASE.jar

**Description:**

Starter for building web, including RESTful, applications using Spring MVC. Uses Tomcat as the default embedded container

**Published Vulnerabilities:**

CVE-2022-27772 - \*\* UNSUPPORTED WHEN ASSIGNED \*\* spring-boot versions prior to version v2.2.11.RELEASE was vulnerable to temporary directory hijacking. This vulnerability impacted the org.springframework.boot.web.server.AbstractConfigurableWebServerFactory.createTempDir method. NOTE: This vulnerability only affects products and/or versions that are no longer supported by the maintainer.

**Dependency:**

spring-core-5.2.3.RELEASE.jar

**Description:**

Spring Core

**Published Vulnerabilities:**

CVE-2022-22965 - 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. However, the nature of the vulnerability is more general, and there may be other ways to exploit it.

CVE-2021-22118

CVE-2020-5421

CVE-2022-22950

CVE-2022-22971

CVE-2022-22968

CVE-2022-22970

CVE-2021-22060

CVE-2021-22096

**Dependency:**

spring-web-5.2.3.RELEASE.jar

**Description:**

Spring Web

**Published Vulnerabilities:**

CVE-2016-1000027 - Pivotal Spring Framework through 5.3.16 suffers from a potential remote code execution (RCE) issue if used for Java deserialization of untrusted data. Depending on how the library is implemented within a product, this issue may or not occur, and authentication may be required. NOTE: the vendor's position is that untrusted data is not an intended use case. The product's behavior will not be changed because some users rely on deserialization of trusted data.

CVE-2022-22965

CVE-2021-22118

CVE-2020-5421

CVE-2022-22950

CVE-2022-22971

CVE-2022-22968

CVE-2022-22970

CVE-2021-22060

CVE-2021-22096

**Dependency:**

spring-webmvc-5.2.3.RELEASE.jar

**Description:**

Spring Web MVC

**Published Vulnerabilities:**

CVE-2022-22965 - 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. However, the nature of the vulnerability is more general, and there may be other ways to exploit it.

CVE-2021-22118

CVE-2020-5421

CVE-2022-22950

CVE-2022-22971

CVE-2022-22968

CVE-2022-22970

CVE-2021-22060

CVE-2021-22096

**Dependency:**

tomcat-embed-core-9.0.30.jar

**Description:**

Core Tomcat implementation

**Published Vulnerabilities:**

CVE-2020-1938 - 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. In Apache Tomcat 9.0.0.M1 to 9.0.0.30, 8.5.0 to 8.5.50 and 7.0.0 to 7.0.99, Tomcat shipped with an AJP Connector enabled by default that listened on all configured IP addresses. It was expected (and recommended in the security guide) that this Connector would be disabled if not required. This vulnerability report identified a mechanism that allowed: - returning arbitrary files from anywhere in the web application - processing any file in the web application as a JSP Further, if the web application allowed file upload and stored those files within the web application (or the attacker was able to control the content of the web application by some other means) then this, along with the ability to process a file as a JSP, made remote code execution possible. It is important to note that mitigation is only required if an AJP port is accessible to untrusted users. Users wishing to take a defence-in-depth approach and block the vector that permits returning arbitrary files and execution as JSP may upgrade to Apache Tomcat 9.0.31, 8.5.51 or 7.0.100 or later. A number of changes were made to the default AJP Connector configuration in 9.0.31 to harden the default configuration. It is likely that users upgrading to 9.0.31, 8.5.51 or 7.0.100 or later will need to make small changes to their configurations.

CVE-2020-11996

CVE-2020-13934

CVE-2020-13935

CVE-2020-17527

CVE-2021-25122

CVE-2021-41079

CVE-2022-29885

CVE-2022-42252

CVE-2020-9484

CVE-2021-25329

CVE-2021-30640

CVE-2022-34305

CVE-2021-24122

CVE-2021-33037

CVE-2019-17569

CVE-2020-1935

CVE-2020-13943

CVE-2021-43980

**Dependency:**

tomcat-embed-websocket-9.0.30.jar

**Description:**

Core Tomcat implementation

**Published Vulnerabilities:**

CVE-2020-1938 - 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. In Apache Tomcat 9.0.0.M1 to 9.0.0.30, 8.5.0 to 8.5.50 and 7.0.0 to 7.0.99, Tomcat shipped with an AJP Connector enabled by default that listened on all configured IP addresses. It was expected (and recommended in the security guide) that this Connector would be disabled if not required. This vulnerability report identified a mechanism that allowed: - returning arbitrary files from anywhere in the web application - processing any file in the web application as a JSP Further, if the web application allowed file upload and stored those files within the web application (or the attacker was able to control the content of the web application by some other means) then this, along with the ability to process a file as a JSP, made remote code execution possible. It is important to note that mitigation is only required if an AJP port is accessible to untrusted users. Users wishing to take a defence-in-depth approach and block the vector that permits returning arbitrary files and execution as JSP may upgrade to Apache Tomcat 9.0.31, 8.5.51 or 7.0.100 or later. A number of changes were made to the default AJP Connector configuration in 9.0.31 to harden the default configuration. It is likely that users upgrading to 9.0.31, 8.5.51 or 7.0.100 or later will need to make small changes to their configurations.

CVE-2020-8022

CVE-2020-11996

CVE-2020-13934

CVE-2020-13935

CVE-2020-17527

CVE-2021-25122

CVE-2021-41079

CVE-2022-29885

CVE-2022-42252

CVE-2020-9484

CVE-2021-25329

CVE-2021-30640

CVE-2022-34305

CVE-2021-24122

CVE-2021-33037

CVE-2019-17569

CVE-2020-1935

CVE-2020-13943

CVE-2021-43980

## Analyze Results

The biggest issues are an old Spring framework and an old Apache Tomcat server. These out-of-date packages account for 69 of the 83 vulnerabilities found. Some other vulnerabilities come from the Maven import and are not being used by the application. They could be considered false positives but it is important to investigate and ensure their presence does not introduce a vulnerability to the system. It may make sense to suppress them for future scans.

It is possible to easily mitigate CVE-2020-10693 by passing the input as an expression variable to HibernateConstraintValidatorContext.

For CVE-2020-25649, it is possible to block expansion and reduce the potential vulnerability exposure however, this action does not fully overcome the vulnerability.

A new version of Log4j (2.17.2) does not have the vulnerability. Upgrading would stop the detection in the dependency check. As we are not using this logging feature in our application, there is little risk but it is also possible to correct the secure SMTP address to stop the vulnerability.

CVE-2021-42550 requires an upgrade to 1.4.5 to resolve this vulnerability.

The snakeyaml vulnerability (CVE-2017-18640) would have minimal impact as it is inside the Srping framework and we are not using the functionality. Leveraging the constructor flaw would result in no code execution. There is no fix in the form of an update so the best mitigation is to only accept connections from known yaml sources.