|  |  |
| --- | --- |
|  |  |

CS 305 Module Two Static Testing Summary Assignment  
Brandon Coulter  
Southern New Hampshire University

# CS 305 Module Two Static Testing Summary Template

### Run Dependency Check

* Running the dependency check on the code base created a HTML report of which a given screen shot is below:Graphical user interface, text, application, email

  Description automatically generated

### Document Results

* Dependency: hibernate-validator-6.0.18.Final.jar
  + CVE Code: [**CVE-2020-10693**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-10693)
  + Description: In version 6.1.2.FINAL of Hibernate Validator, the process of interpolating Expression Language expressions could assess invalid expression to be valid allowing attacks based on bypassed input validation due to the fact that they were erroneously validated. (Booth et. al., 2015)
* Dependency: Jackson-databind-2.10.2.jar
  + CVE Code: [**CVE-2020-25649**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-25649)
  + Description: This vulnerability has the risk of compromising data integrity due to XML external entity (XXE) attacks. This is due to a lack in entity expansion. (Booth et. al., 2015)
  + CVE Code: [**CVE-2020-36518**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-36518)
  + Description: Due to a large amount of nested object, attacker could possible conduct a denial of service attack. (Booth et. al., 2015)
* Dependency: log4j-api-2.12.1.jar
  + CVE Code: [**CVE-2020-9488**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-9488)
  + Description: A flaw in certificate validation allowed man-in-the-middle attacks to occur potentially breaching log messages sent through the mismatched appender. (Booth et. al., 2015)
* Dependency: logback-core-1.2.3.jar
  + CVE Code: [**CVE-2021-42550**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-42550)
  + Description: In versions 1.2.7 or any version prior, attackers could utilize required privileges to execute capricious code stored on a LDAP server by editing files to manufacture malicious configurations. (Booth et. al., 2015)
* Dependency: mongo-java-driver-2.4.jar
  + CVE Code: **CVE-2021-20328**
  + Description: This vulnerability allowed for an active man-in-the-middle attack to capture data between the java driver and key management service(KMS) effectively making encryption unsuccessful as the data is intercepted in the middle. This is caused by a failure to verify the host name of a KMS server as well as a privileged network position being used. (Booth et. al., 2015)
* Dependency: snakeyami-1.25.jar
  + CVE Code: [**CVE-2017-18640**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2017-18640)
  + Description: In 1.18 version of SnakeYAML, the alias feature would permit entity expansion. This vulnerability allows for a denial of service attack to be performed by endless entity expansion. (Booth et. al., 2015)
* Dependency: spring-boot-2.2.4.RELEASE.jar
  + CVE Code: [**CVE-2022-27772**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-27772)
  + Description: Vulnerability in this version and all versions prior to 2.2.11.RELEASE could potentially allow attackers to seize the directory. (Booth et. al., 2015)
* Dependency: spring-core-5.2.3.RELEASE.jar
  + CVE Code: [**CVE-2022-22965**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-22965)
  + Description: Utilizing the technique data binding, attackers could execute code remotely through Spring WebFlux application that are running version 9+ of Java’s JDK. (Booth et. al., 2015)
  + CVE Code: [**CVE-2021-22118**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-22118)
  + Description: An attacker could read or modify files on a WebFlux application due to versions 5.2.15 and prior versions on the software being vulnerable to privilege escalation. (Booth et. al., 2015)
  + CVE Code: [**CVE-2020-5421**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-5421)
  + Description: Attacks were able to bypass RFD attack protections from earlier versions. (Booth et. al., 2015)
  + CVE Code: [**CVE-2022-22950**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-22950)
  + Description: An attacker could potentially create a denial of service situation by inserting a special SpEL expression. (Booth et. al., 2015)
  + CVE Code: [**CVE-2022-22971**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-22971)
  + Description: If an application has Simply text-oriented message protocol over WebSocket endpoints, it could be potentially be vulnerable to a denial of service attack conducted by an authenticated user. This is possible on versions 5.3.20+ and 5.2.22+ and all unsupported versions prior. (Booth et. al., 2015)
  + CVE Code: [**CVE-2022-22968**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-22968)
  + Description: This version is not properly protected for disallowedFields on a DataBinder due to lack of cases sensitivity. (Booth et. al., 2015)
  + CVE Code: [**CVE-2022-22970**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-22970)
  + Description: Vulnerable to denial of service attacks, files uploads that rely on data binding could potentially come under attack. (Booth et. al., 2015)
  + CVE Code: [**CVE-2021-22060**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-22060)
  + Description: Malicious users can potentially provide input that could lead to insertion of extra log records (Booth et. al., 2015)
  + CVE Code: [**CVE-2021-22096**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-22096)
  + Description: Utilizing malicious inputs, an attacker could create additional log entries. (Booth et. al., 2015)
* Dependency: spring-web-5.2.3.RELEASE.jar
  + CVE Code: [**CVE-2016-1000027**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2016-1000027)
  + Description: Using java deserialization for untrusted information, spring framework 5.3.16 could allow remote code execution. (Booth et. al., 2015)
  + CVE Code: [**CVE-2022-22965**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-22965)
  + Description Utilizing the technique data binding, attackers could execute code remotely through Spring WebFlux application that are running version 9+ of Java’s JDK. (Booth et. al., 2015)
  + CVE Code: [**CVE-2021-22118**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-22118)
  + Description: An attacker could read or modify files on a WebFlux application due to versions 5.2.15 and prior versions on the software being vulnerable to privilege escalation. (Booth et. al., 2015)
  + CVE Code: [**CVE-2020-5421**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-5421)
  + Description: Attacks were able to bypass RFD attack protections from earlier versions. (Booth et. al., 2015)
  + CVE Code: [**CVE-2022-22950**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-22950)
  + Description: An attacker could potentially create a denial of service situation by inserting a special SpEL expression. (Booth et. al., 2015)
  + CVE Code: [**CVE-2022-22971**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-22971)
  + Description: If an application has Simply text-oriented message protocol over WebSocket endpoints, it could be potentially be vulnerable to a denial of service attack conducted by an authenticated user. This is possible on versions 5.3.20+ and 5.2.22+ and all unsupported versions prior. (Booth et. al., 2015)
  + CVE Code: [**CVE-2022-22968**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-22968)
  + Description: This version is not properly protected for disallowedFields on a DataBinder due to lack of cases sensitivity. (Booth et. al., 2015)
  + CVE Code: [**CVE-2022-22970**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-22970)
  + Description: Vulnerable to denial of service attacks, files uploads that rely on data binding could potentially come under attack. (Booth et. al., 2015)
  + CVE Code: [**CVE-2021-22060**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-22060)
  + Description: Malicious users can potentially provide input that could lead to insertion of extra log records. (Booth et. al., 2015)
  + CVE Code: [**CVE-2021-22096**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-22096)
  + Description: Utilizing malicious inputs, an attacker could create additional log entries. (Booth et. al., 2015)
* Dependency: tomcat-embed-core-9.0.30.jar
  + CVE Code: [**CVE-2020-1938**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-1938)
  + Description: If an application is using Apache JServ protocol it is critical to be weary of inbound connections to Apache Tomcat due to an attacker’s ability to potentially execute code remotely. (Booth et. al., 2015)
  + CVE Code:[**CVE-2020-11996**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-11996)
  + Description: A denial of service attack could conceivably occur if an attacker used specialize sequences of HTTP/2 requests to overload the servers CPU. (Booth et. al., 2015)
  + CVE Code:[**CVE-2020-13934**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-13934)
  + Description: A memory leak could occur causing a denial of service due to a ample number of request being made by an attacker and the processor not being released. (Booth et. al., 2015)
  + CVE Code:[**CVE-2020-13935**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-13935)
  + Description: This version of Apache Tomcat did not validate the payload length in a WebSocket frame and thus an infinite loop could be initiated causing a denial of service. (Booth et. al., 2015)
  + CVE Code: [**CVE-2020-17527**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-17527)
  + Description: A possibility of information leaking could happen between multiple requests if the request header value of an HTTP request was reused from the prior stream obtained. (Booth et. al., 2015)
  + CVE Code:[**CVE-2021-25122**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-25122)
  + Description: This version of Apache Tomcat potentially duplicated request headers and partial body, thus allowing a user to see a request generated by another user. (Booth et. al., 2015)
  + CVE Code:[**CVE-2021-41079**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-41079)
  + Description: A denial of service could be caused by a specialized package that caused an infinite loop do to none validated inbound TLS packets. (Booth et. al., 2015)
  + CVE Code:[**CVE-2022-29885**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-29885)
  + Description: Apace Tomcat documentation misguided users on information for the Encryptinterceptor. (Booth et. al., 2015)
  + CVE Code:[**CVE-2020-9484**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-9484)
  + Description: Given a few preexisting conditions are met and attacker could potentially activate a remote code execution through deserialization of files they hold control over. (Booth et. al., 2015)
  + CVE Code:[**CVE-2021-25329**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-25329)
  + Description: Although improbable, a given edge case configuration could lead to a remote code execution similar to CVE-2020-9484. (Booth et. al., 2015)
  + CVE Code:[**CVE-2021-30640**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-30640)
  + Description: This vulnerability allows outside attackers to authenticate a log in using variants of valid usernames in the JNDI portion of Apache. (Booth et. al., 2015)
  + CVE Code:[**CVE-2022-34305**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-34305)
  + Description: Apache Tomcat of this version provided a means for cross site scripting (XSS) buy not filtering exposed user supplied information. (Booth et. al., 2015)
  + CVE Code:[**CVE-2021-24122**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-24122)
  + Description: Due to certain configurations, this Apache Tomcat version was predisposed to JSP source code leak when passing assets from a given network utilizing the NTFS file system. (Booth et. al., 2015)
  + CVE Code:[**CVE-2021-33037**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-33037)
  + Description: By not parsing a HTTP request header correctly, Apache Tomcat had the possibility to become vulnerable to request smuggling. (Booth et. al., 2015)
  + CVE Code: [**CVE-2019-17569**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2019-17569)
  + Description: Request smuggling was possible due to regression which resulted in non-valid transfer-encoding headers that were improperly handled. This was caused by a refactor of the previous version of Apache Tomcat. (Booth et. al., 2015)
  + CVE Code:[**CVE-2020-1935**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-1935)
  + Description: A parsing error in the code that was used to parse HTTP headers allowed some parsing to pass as valid whilst not truly valid resulting in a potential HTTP request smuggling. (Booth et. al., 2015)
  + CVE Code:[**CVE-2020-13943**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-13943)
  + Description: Violating HTTP/2 protocols, a client connecting to Apache Tomcat in the given version could surpass the limit for simultaneous streams resulting in possible erroneous responses. (Booth et. al., 2015)
* Dependency: tomcat-embed-websocket-9.0.30.jar
  + CVE Code:[**CVE-2020-1938**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-1938)
  + Description: If an application is using Apache JServ protocol it is critical to be weary of inbound connections to Apache Tomcat due to an attacker’s ability to potentially execute code remotely. (Booth et. al., 2015)
  + CVE Code:[**CVE-2020-8022**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-8022)
  + Description: Due to erroneous default permissions, the packaging of Tomcat left room for vulnerabilities. (Booth et. al., 2015)
  + CVE Code:[**CVE-2020-11996**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-11996)
  + Description: A denial of service attack could conceivably occur if an attacker used specialize sequences of HTTP/2 requests to overload the servers CPU. (Booth et. al., 2015)
  + CVE Code:[**CVE-2020-13934**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-13934)
  + Description: A memory leak could occur causing a denial of service due to a ample number of request being made by an attacker and the processor not being released. (Booth et. al., 2015)
  + CVE Code:[**CVE-2020-13935**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-13935)
  + Description: This version of Apache Tomcat did not validate the payload length in a WebSocket frame and thus an infinite loop could be initiated causing a denial of service. (Booth et. al., 2015)
  + CVE Code:[**CVE-2021-25122**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-25122)
  + Description: This version of Apache Tomcat potentially duplicated request headers and partial body, thus allowing a user to see a request generated by another user. (Booth et. al., 2015)
  + CVE Code:[**CVE-2021-41079**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-41079)
  + Description: A denial of service could be caused by a specialized package that caused an infinite loop do to none validated inbound TLS packets. (Booth et. al., 2015)
  + CVE Code:[**CVE-2022-29885**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-29885)
  + Description: Apace Tomcat documentation misguided users on information for the Encryptinterceptor. (Booth et. al., 2015)
  + CVE Code:[**CVE-2020-9484**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-9484)
  + Description: Given a few preexisting conditions are met and attacker could potentially activate a remote code execution through deserialization of files they hold control over. (Booth et. al., 2015)
  + CVE Code:[**CVE-2021-25329**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-25329)
  + Description: Although improbable, a given edge case configuration could lead to a remote code execution similar to CVE-2020-9484 (Booth et. al., 2015)
  + CVE Code:[**CVE-2021-30640**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-30640)
  + Description: This vulnerability allows outside attackers to authenticate a log in using variants of valid usernames in the JNDI portion of Apache. (Booth et. al., 2015)
  + CVE Code:[**CVE-2022-34305**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2022-34305)
  + Description: Apache Tomcat of this version provided a means for cross site scripting (XSS) buy not filtering exposed user supplied information. (Booth et. al., 2015)
  + CVE Code:[**CVE-2021-24122**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-24122)
  + Description: Due to certain configurations, this Apache Tomcat version was predisposed to JSP source code leak when passing assets from a given network utilizing the NTFS file system. (Booth et. al., 2015)
  + CVE Code:[**CVE-2021-33037**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2021-33037)
  + Description: By not parsing a HTTP request header correctly, Apache Tomcat had the possibility to become vulnerable to request smuggling. (Booth et. al., 2015)
  + CVE Code: [**CVE-2019-17569**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2019-17569)
  + Description: Request smuggling was possible due to regression which resulted in non-valid transfer-encoding headers that were improperly handled. This was caused by a refactor of the previous version of Apache Tomcat. (Booth et. al., 2015)
  + CVE Code:[**CVE-2020-1935**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-1935)
  + Description: A parsing error in the code that was used to parse HTTP headers allowed some parsing to pass as valid whilst not truly valid resulting in a potential HTTP request smuggling. (Booth et. al., 2015)
  + CVE Code:[**CVE-2020-13943**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-13943)
  + Description: Violating HTTP/2 protocols, a client connecting to Apache Tomcat in the given version could surpass the limit for simultaneous streams resulting in possible erroneous responses. (Booth et. al., 2015)

### Interpret Results

* Based on the hits from the dependency check, the solutions for a majority of the vulnerabilities can be fixed by updating the dependency libraries. As most of the dependencies are out of date and older versions of the given dependency, simply updating the dependencies will mitigate their vulnerabilities. As these susceptibilities are found and documented, solutions, fixes and patches are created and released for use. Not to say those fixes cannot be exploited in new versions in alternative ways, but a majority of the weaknesses are corrected in later versions of each of the dependency libraries. As for the false positives of the dependency check report, it is important to filter out and remove reports that are false positive. That does not mean they are not worth looking into as a baseline for the security of your application. However, by removing false positives, it clears the way for more pressing vulnerabilities, easing the process for conducting the vulnerability check.

**References**

Booth, H., Rike, D. and Witte, G. (2013), The National Vulnerability Database (NVD): Overview, ITL Bulletin, National Institute of Standards and Technology, Gaithersburg, MD, [online], https://tsapps.nist.gov/publication/get\_pdf.cfm?pub\_id=915172 (Accessed July 10, 2022)