CS 305 Software Security  
Static Testing

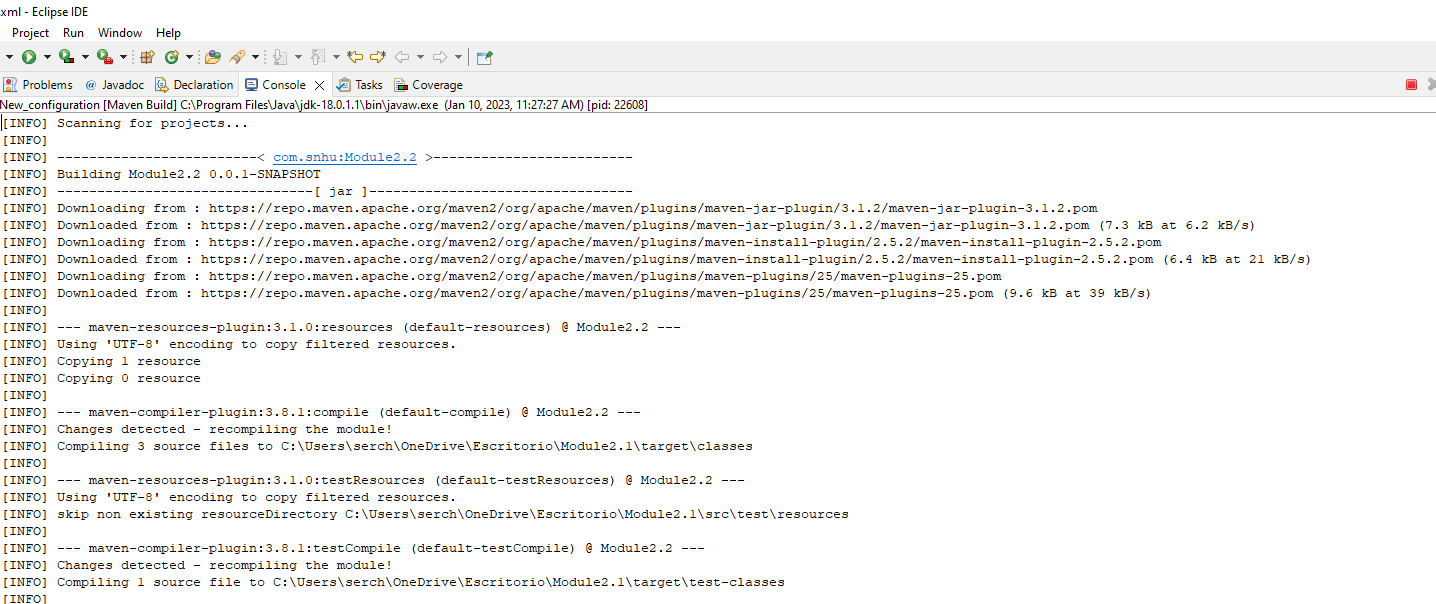
Sergio Mateos

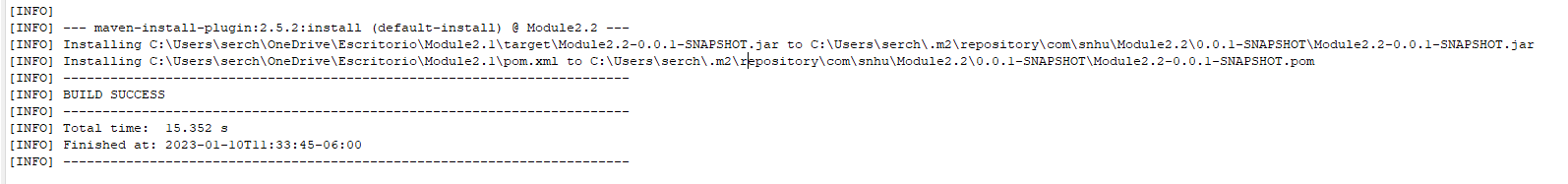
Southern New Hampshire University

Static Testing

**Run Dependency Check**

Maven Built





A picture containing table

Description automatically generated

**Document Results**

Graphical user interface, text

Description automatically generated

**Interpret Result**

* False-Positive: It’s common that while running the dependency check may discover false-positives, this may not indicate severe problems, but it needs to explore to prevent any security risk. Exploring the false-positives may guide you into bigger problems and security risks.
* [hibernate-validator-6.0.18.Final.jar](file:///C:\Users\serch\OneDrive\Escritorio\Module2.1\target\dependency-check-report.html#l2_7fd00bcd87e14b6ba66279282ef15efa30dd2492): 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. This means enable invalid El expressions and they are evaluated as valid.
* [jackson-databind-2.10.2.jar](file:///C:\Users\serch\OneDrive\Escritorio\Module2.1\target\dependency-check-report.html#l4_0528de95f198afafbcfb0c09d2e43b6e0ea663ec): 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. This means data an be compromise by having access from external entities to the system data and that way have access to the information.
* [log4j-api-2.12.1.jar](file:///C:\Users\serch\OneDrive\Escritorio\Module2.1\target\dependency-check-report.html#l9_a55e6d987f50a515c9260b0451b4fa217dc539cb): 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. Description: This means a malicious user can access to our network by the securety fail of the connection.
* [logback-core-1.2.3.jar](file:///C:\Users\serch\OneDrive\Escritorio\Module2.1\target\dependency-check-report.html#l11_864344400c3d4d92dfeb0a305dc87d953677c03c): 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. Description: This means malicious user who has access to our system have edit, insert, and delete files that can produce a failure in the system.
* [snakeyaml-1.25.jar](file:///C:\Users\serch\OneDrive\Escritorio\Module2.1\target\dependency-check-report.html#l14_8b6e01ef661d8378ae6dd7b511a7f2a33fae1421): 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. We recommend using SnakeYaml's SafeConsturctor when parsing untrusted content to restrict deserialization. Description: This means that the malicious user can access remotely because the Contrustor() class does not restrict and can pass untrusted inputs.
* [spring-boot-2.2.4.RELEASE.jar](file:///C:\Users\serch\OneDrive\Escritorio\Module2.1\target\dependency-check-report.html#l15_225a4fd31156c254e3bb92adb42ee8c6de812714): \*\* 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. Description: This means that the system is expose to hijacking and the vulnerability of an attack increase.
* [spring-core-5.2.3.RELEASE.jar](file:///C:\Users\serch\OneDrive\Escritorio\Module2.1\target\dependency-check-report.html#l17_3734223040040e8c3fecd5faa3ae8a1ed6da146b): 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. Description: The system is vulnerable to remote code execution and it need to be found the specific exploit to avoid the vulnerability.
* [tomcat-embed-core-9.0.30.jar](file:///C:\Users\serch\OneDrive\Escritorio\Module2.1\target\dependency-check-report.html#l20_ad32909314fe2ba02cec036434c0addd19bcc580) : Error: 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 defense-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. Description: There were couple errors found which led to an instability on the memory.
* [tomcat-embed-websocket-9.0.30.jar](file:///C:\Users\serch\OneDrive\Escritorio\Module2.1\target\dependency-check-report.html#l22_33157f6bc5bfd03380ebb5ac476db0600a04168d): 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. Description: This means the parse was incorrect and this transfer encoding request. The transfer of encoding request can lead to an anonymous access.

Fist the system has always has to have the latest version to prevent any exposure from malicious users. Validation verification of inputs, privilege management, and certification handling can prevent vulnerability in the system. Dependencies must be verified to prevent any issue or replacement. False-Positive must be inspected to avoid any scalation of the issue, or analyzing possible issues connected to the problem.