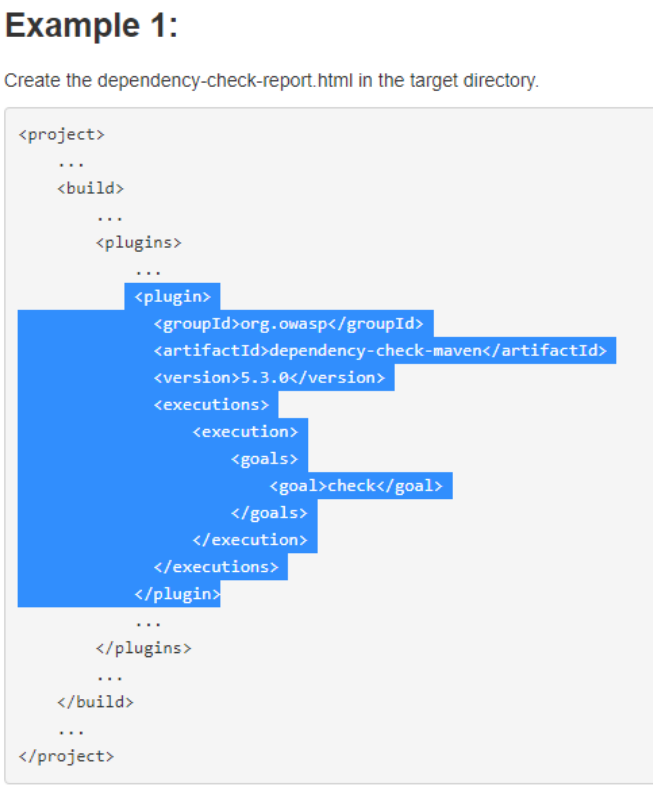
Southern
New Hampshire
University

CS 305 Integrating the Maven Dependency-Check Plug-In Tutorial

This course uses Maven, which is a package manager. You can run the dependency check as a standalone application or as part of a Maven project. This tutorial shows you how to run a dependency check as a Maven project. Follow these steps to integrate the dependency-check plug-in.

Note: You should run a Hello World program successfully before modifying your pom.xml file.

1. Go to the OWASP Dependency-Check Maven webpage linked in the Module Three and Seven Resources sections. Refer to the example provided on the webpage and as shown below.

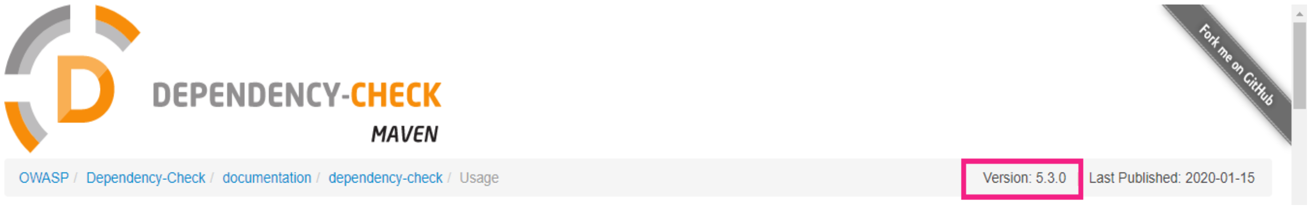


Example 1:

Create the dependency-check-report.html in the target directory.

```
<project>
    ...
    <build>
        ...
        <plugins>
            ...
            <plugin>
              <groupId>org.owasp</groupId>
              <artifactId>dependency-check-maven</artifactId>
              <version>5.3.0</version>
              <executions>
                  <execution>
                      <goals>
                          <goal>check</goal>
                      </goals>
                  </execution>
              </executions>
            </plugin>
            ...
        </plugins>
        ...
    </build>
    ...
</project>
```

2. Identify the Dependency-Check version number. You will need this information for step 3.


DEPENDENCY-CHECK
MAVEN
Fork me on GitHub
OWASP / Dependency-Check / documentation / dependency-check / Usage

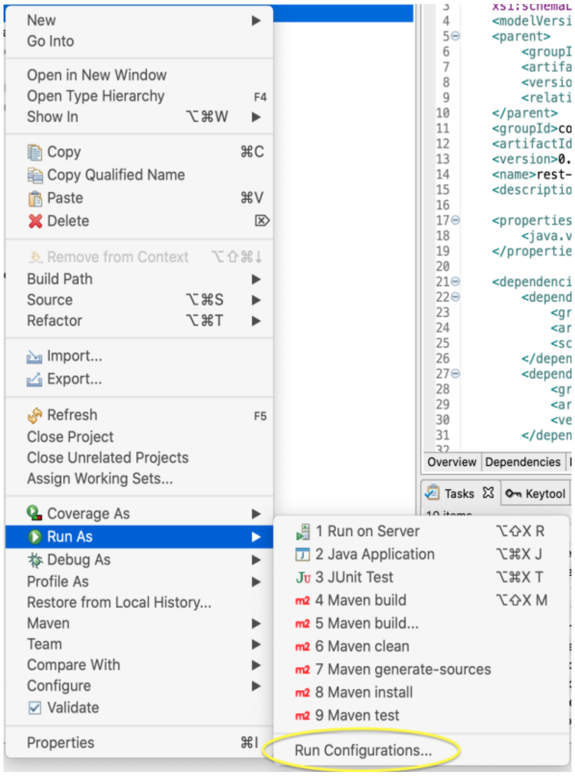
Version: 5.3.0 Last Published: 2020-01-15

3. Next, upload the Maven project you would like to complete a static test for in Eclipse.
    A. Open the pom.xml file to add the dependency-check plug-in.
    B. Copy the following lines of code and paste the code into the pom.xml file. Be certain you're using the latest version based on your findings in step 2.

```
<plugin>
   <groupId>org.owasp</groupId>
   <artifactId>dependency-check-maven</artifactId>
   <version>9.0.7</version>
   <executions>
      <execution>
         <goals>
            <goal>check</goal>
         </goals>
      </execution>
   </executions>
</plugin>
```

Note: You will need to complete these steps for **each** code base or software application to use the dependency-check plug-in.

4. Eclipse will run the Maven plug-in when you compile your code. Run the pom.xml file to make certain that the plug-in is running effectively.
    A. On the menu bar, click **Run**, **Run As**, and **Run Configurations.**

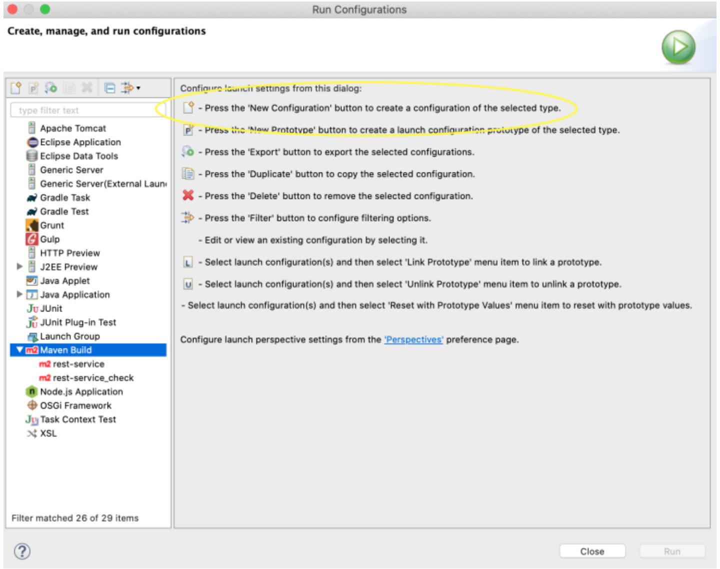


```
xsi:schemaL
<modelVersi
<parent>
    <groupI
    <artifa
    <versio
    <relati
</parent>
<groupId>co
<artifactId
<version>0.
<name>rest-
<descriptio

<properties
    <java.v
</propertie

<dependenci
    <depend
        <gr
        <ar
        <sc
    </depen
    <depend
        <gr
        <ar
        <ve
    </depen
```

    B. Double-click **Maven Build** in the Run Configurations window in Eclipse.



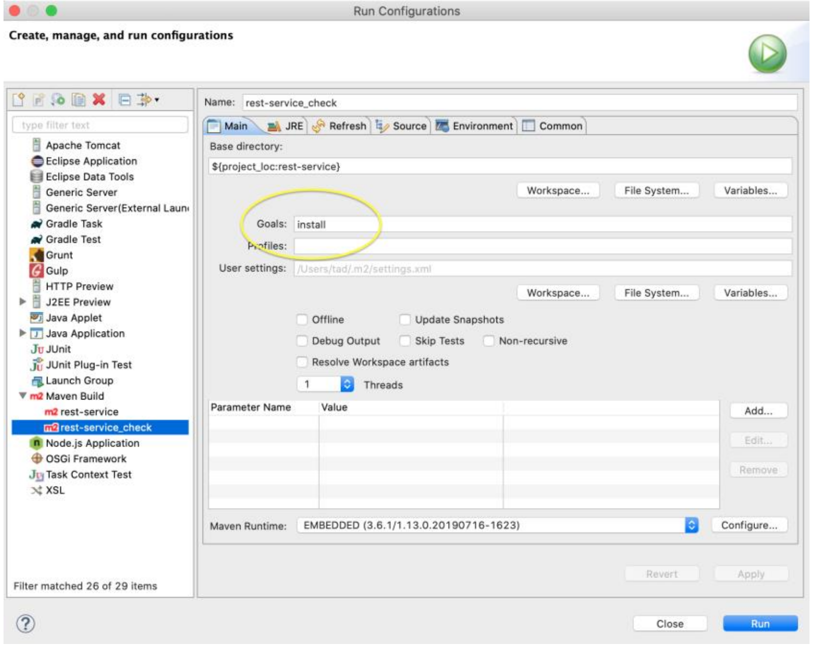
Configure launch settings from this dialog:

- Press the 'New Configuration' button to create a configuration of the selected type.
- Press the 'New Prototype' button to create a launch configuration prototype of the selected type.
- Press the 'Export' button to export the selected configurations.
- Press the 'Duplicate' button to copy the selected configuration.
- Press the 'Delete' button to remove the selected configuration.
- Press the 'Filter' button to configure filtering options.
- Edit or view an existing configuration by selecting it.
- Select launch configuration(s) and then select 'Link Prototype' menu item to link a prototype.
- Select launch configuration(s) and then select 'Unlink Prototype' menu item to unlink a prototype.
- Select launch configuration(s) and then select 'Reset with Prototype Values' menu item to reset with prototype values.

Configure launch perspective settings from the 'Perspectives' preference page.

i. Choose the Base directory if it is not already present by clicking **Workspace.** Then select the appropriate directory.

C. Enter “install” in the **Goals** field. Then click **Run**.



Name: rest-service_check

Base directory: ${project_loc:rest-service}

Goals: install

Profiles:

User settings: /Users/tad/.m2/settings.xml

□ Offline □ Update Snapshots

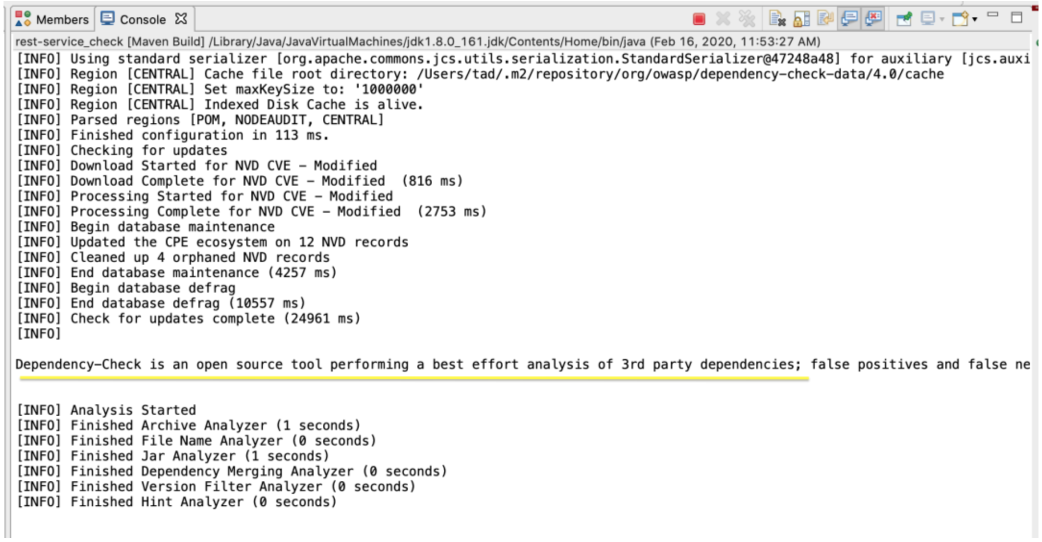
□ Debug Output □ Skip Tests □ Non-recursive

□ Resolve Workspace artifacts

1 Threads

Maven Runtime: EMBEDDED (3.6.1/1.13.0.20190716-1623)

D. Be certain to observe the **Console** for dependency-check execution. The first time you do this, it will require more time to download the Common Vulnerabilities and Exposures (CVE) from the National Vulnerabilities Database (NVD).



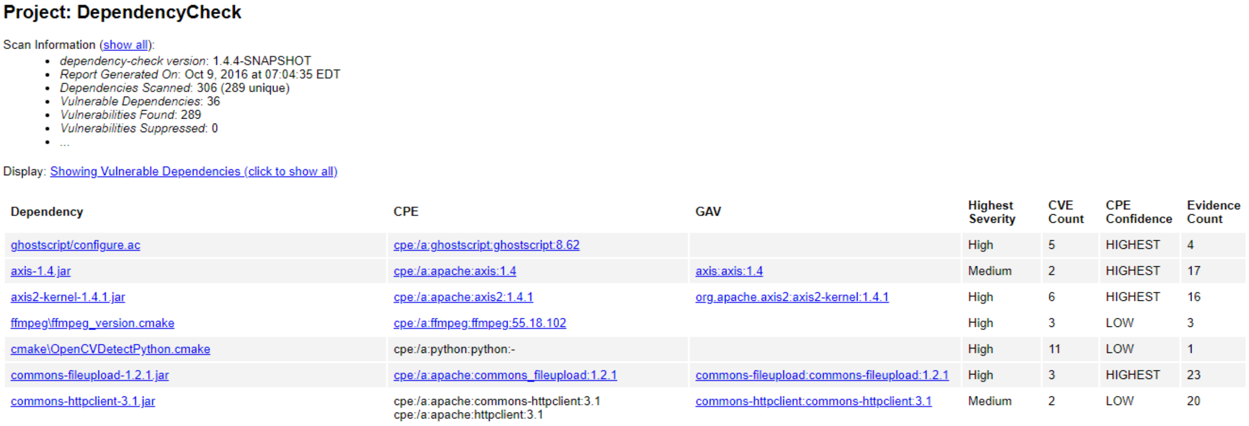
```
rest-service_check [Maven Build] /Library/Java/JavaVirtualMachines/jdk1.8.0_161.jdk/Contents/Home/bin/java (Feb 16, 2020, 11:53:27 AM)
[INFO] Using standard serializer [org.apache.commons.jcs.utils.serialization.StandardSerializer@47248a48] for auxiliary [jcs.auxi
[INFO] Region [CENTRAL] Cache file root directory: /Users/tad/.m2/repository/org/owasp/dependency-check-data/4.0/cache
[INFO] Region [CENTRAL] Set maxKeySize to: '1000000'
[INFO] Region [CENTRAL] Indexed Disk Cache is alive.
[INFO] Parsed regions [POM, NODEAUDIT, CENTRAL]
[INFO] Finished configuration in 113 ms.
[INFO] Checking for updates
[INFO] Download Started for NVD CVE - Modified
[INFO] Download Complete for NVD CVE - Modified  (816 ms)
[INFO] Processing Started for NVD CVE - Modified
[INFO] Processing Complete for NVD CVE - Modified  (2753 ms)
[INFO] Begin database maintenance
[INFO] Updated the CPE ecosystem on 12 NVD records
[INFO] Cleaned up 4 orphaned NVD records
[INFO] End database maintenance (4257 ms)
[INFO] Begin database defrag
[INFO] End database defrag (10557 ms)
[INFO] Check for updates complete (24961 ms)
[INFO]

Dependency-Check is an open source tool performing a best effort analysis of 3rd party dependencies; false positives and false ne


[INFO] Analysis Started
[INFO] Finished Archive Analyzer (1 seconds)
[INFO] Finished File Name Analyzer (0 seconds)
[INFO] Finished Jar Analyzer (1 seconds)
[INFO] Finished Dependency Merging Analyzer (0 seconds)
[INFO] Finished Version Filter Analyzer (0 seconds)
[INFO] Finished Hint Analyzer (0 seconds)
```

E. When the Run action is complete, check under the **Target director** of the project to see the **dependency-check-report.html** report.

   i. A sample report is available at Project: DependencyCheck. The output report should look like the example below.



Project: DependencyCheck

Scan Information (show all):

- *dependency-check version*: 1.4.4-SNAPSHOT
- *Report Generated On*: Oct 9, 2016 at 07:04:35 EDT
- *Dependencies Scanned*: 306 (289 unique)
- *Vulnerable Dependencies*: 36
- *Vulnerabilities Found*: 289
- *Vulnerabilities Suppressed*: 0
- ...

Display: Showing Vulnerable Dependencies (click to show all)

Dependency	CPE	GAV	Highest Severity	CVE Count	CPE Confidence	Evidence Count
ghostscript/configure.ac	cpe:/a:ghostscript:ghostscript:8.62		High	5	HIGHEST	4
axis-1.4.jar	cpe:/a:apache:axis:1.4	axis:axis:1.4	Medium	2	HIGHEST	17
axis2-kernel-1.4.1.jar	cpe:/a:apache:axis2:1.4.1	org.apache.axis2:axis2-kernel:1.4.1	High	6	HIGHEST	16
ffmpeg\ffmpeg_version.cmake	cpe:/a:ffmpeg:ffmpeg:55.18.102		High	3	LOW	3
cmake\OpenCVDetectPython.cmake	cpe:/a:python:python:-		High	11	LOW	1
commons-fileupload-1.2.1.jar	cpe:/a:apache:commons_fileupload:1.2.1	commons-fileupload:commons-fileupload:1.2.1	High	3	HIGHEST	23
commons-httpclient-3.1.jar	cpe:/a:apache:commons-httpclient:3.1 cpe:/a:apache:httpclient:3.1	commons-httpclient:commons-httpclient:3.1	Medium	2	LOW	20