OWASP Dependency Check

Dependency-Check is a software composition analysis tool that helps identify known vulnerabilities in project dependencies. To install and execute Dependency-Check on Linux Ubuntu, follow these steps:

Step 1: Install Java Development Kit (JDK)

Dependency-Check requires Java to run. If you don't have Java installed, you can install it using the following commands:

sudo apt update

sudo apt install default-jdk

Step 2: Download Dependency-Check

You can download Dependency-Check from the official GitHub repository. You can use the following commands to download the tool:

wget https://github.com/jeremylong/DependencyCheck/releases/download/v6.2.1/dependency-check-6.2.1-release.zip

Make sure to replace the URL with the latest release URL if there's a newer version available.

Step 3: Unzip the Archive

Unzip the downloaded archive using the following command:

unzip dependency-check-6.2.1-release.zip

Again, replace the filename with the appropriate version if needed.

Step 4: Run Dependency-Check

Navigate to the extracted directory and execute Dependency-Check:

cd dependency-check

./bin/dependency-check.sh --project YourProjectName --scan /path/to/your/project

Replace YourProjectName with the desired project name and /path/to/your/project with the actual path to your project's root directory that you want to scan for vulnerabilities.

Dependency-Check will start analyzing the project's dependencies and searching for known vulnerabilities in various databases. Once the scan is complete, you will see a report generated in various formats (HTML, XML, JSON) in the dependency-check-report directory.

Remember that Dependency-Check's effectiveness depends on its database of known vulnerabilities, so make sure to keep it up-to-date by regularly updating the tool.

Integration In Jenkins

To integrate OWASP Dependency Check with Jenkins, you can follow these steps:

1. Install the OWASP Dependency Check plugin in Jenkins. Go to Jenkins Dashboard, click on "Manage Jenkins," then select "Manage Plugins." In the "Available" tab, search for "OWASP Dependency Check" and install the plugin.

2. Configure the OWASP Dependency Check plugin. After installing the plugin, go to the Jenkins Dashboard and click on "Manage Jenkins" again. This time, select "Tools/Global Tool COnfiguration" Scroll down to the "OWASP Dependency Check" section and provide the necessary configuration details, such as the version of Dependency Check **3.** Set up a Jenkins job. Create a new Jenkins pipeline job or open an existing one. In the job configuration, add a stage as below

```
stage('OWASP Dependency Check') {
    steps {
        dependencyCheck additionalArguments: ' --scan ./ ', odcInstallation: 'DC'
        dependencyCheckPublisher pattern: '**/dependency-check-report.xml'
    }
}
```

This code snippet represents a Jenkins pipeline stage named "OWASP Dependency Check." Within the stage, there are two steps:

3.1 The dependencyCheck step is used to execute the OWASP Dependency Check tool. It takes two parameters: additionalArguments specifies any additional command-line arguments you want to pass to the tool, and odcInstallation specifies the installation of the Dependency Check tool in Jenkins.

3.2 The dependencyCheckPublisher step is used to publish the generated Dependency Check report. The pattern parameter specifies the file pattern to search for the report file (dependency-check-report.xml) in the workspace.

This pipeline stage allows you to integrate OWASP Dependency Check into your Jenkins pipeline and generate a report for further analysis.

Save the Jenkins job configuration and run the job. Once you have configured the job, save the configuration, and run the job. Jenkins will execute the OWASP Dependency Check and generate a report.

View the OWASP Dependency Check report. After the job completes, you can view the OWASP Dependency Check report by going to the job's build page and clicking on the "OWASP Dependency Check" link. The report will provide information about any known vulnerabilities or outdated dependencies in your project's dependencies. By integrating OWASP Dependency Check with Jenkins, you can automate the process of scanning your project's dependencies for security vulnerabilities, helping you identify and address potential risks early in the development lifecycle.

Important URL:

https://github.com/jeremylong/DependencyCheck

https://jeremylong.github.io/DependencyCheck/dependency-check-cli/arguments.html

Complete Pipeline

Jenkins Pipeline

```
pipeline {
```

```
agent any
tools {
    jdk 'jdk17'
    maven 'maven3'
}
environment {
    SCANNER_HOME = tool 'sonar-scanner'
}

stages {
    stage('git-checkout') {
        steps {
            git 'https://github.com/jaiswaladi2468/BoardgameListingWebApp.git'
        }
    }

    stage('Code-Compile') {
        steps {
            sh "mvn clean compile"
        }
    }

    stage('Unit-Test') {
        steps {
            sh "mvn clean test"
        }
    }

    stage('OWASP Dependency Check') {
        steps {
            dependencyCheck additionalArguments: ' --scan . ', odcInstallation: 'DC'
```

```
            dependencyCheckPublisher pattern: '**/dependency-check-report.xml'
        }
      }

      stage('Code-Build') {
        steps {
          sh "mvn clean package"
        }
      }

      stage('Sonar Analysis') {
        steps {
          withSonarQubeEnv('sonar') {
            sh ''' $SCANNER_HOME/bin/sonar-scanner -Dsonar.projectName=Devops-CICD \
            -Dsonar.java.binaries=. \
            -Dsonar.projectKey=Devops-CICD '''
          }
        }
      }
    }
}
```