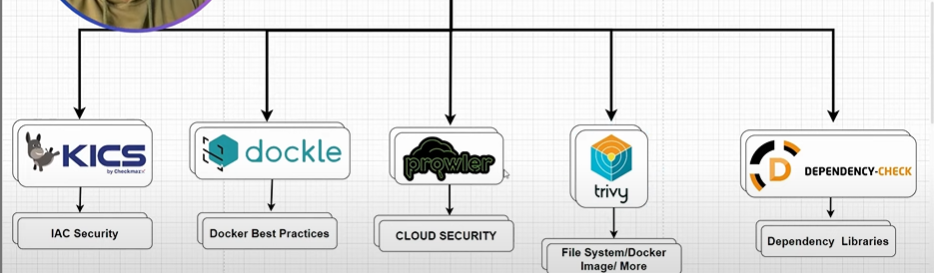
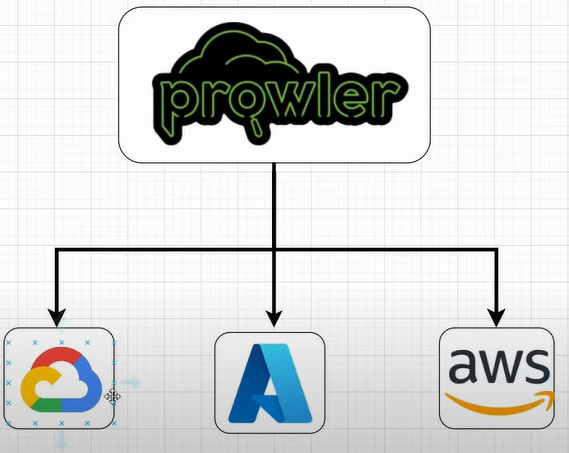
**Refer URL:** <https://www.youtube.com/watch?v=5CbQyB0dcCA&list=PLAdTNzDIZj_8Gotv6qZoyxOiwu0bdhW-E>

**Security Tools**





A screen shot of a computer

Description automatically generated

A screenshot of a computer error

Description automatically generated

**What is Prowler?**

**🡺** Prowler is an open-source security tool designed for AWS (Amazon Web Services) environments. It is used to assess and improve the security posture of AWS accounts by performing compliance checks and identifying potential vulnerabilities.

🡺 CIS benchmarks provide guidance for securely configuring various technologies, and Prowler helps to automate the checks against these benchmarks in cloud environments.

**Usage Examples:**

1. **List Categories and Compliance:**

- prowler aws --list-categories

- prowler aws --list-compliance

1. **Run Prowler Scan:**

- prowler aws

1. **List AWS Services and Run Specific Services:**

- prowler aws --list-services

-prowler aws --services s3 ec2

1. **Exclude Specific Checks:**

- prowler aws --excluded-checks s3\_bucket\_public\_access

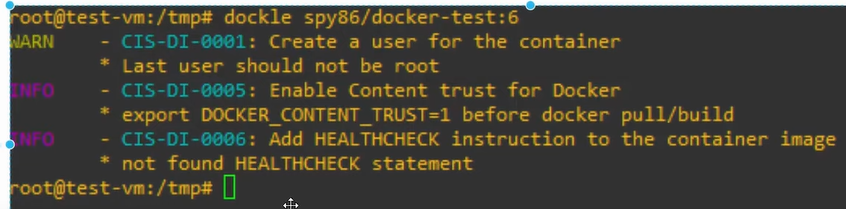
1. **Specify AWS Profile and Region:**

- prowler aws --profile custom-profile -r us-east-1

A blue and white logo

Description automatically generated

Garnet Format:



**CIS ----> Center for Internet security.**

**Example for Dockerfile image:**

**A screenshot of a computer

Description automatically generated**

**A diagram of a diagram

Description automatically generated**

**Note:**

**🡺Two main thinking trivy is going to scan.**

**1.** It going to scan the dependencies that are bring used in our project. When I say dependencies that means for example this it’s a java-based project if I go to POM.Xml file so all specific library that are being used by respect to this project are mentioned in our POM file inside the Tag dependencies. It will scan and compare with specific database.

2. It perform a security scan for the sensitive data inside our source code or inside our folder file system. It’s any like password or token or any secret kind of information mentioned in our it fined out way of doing not good this kind of security issue.

Why trivy is better than OS dependency check because os also can perform same take?

**Installing Step:**

sudo apt-get install wget apt-transport-https gnupg lsb-release

wget -qO - https://aquasecurity.github.io/trivy-repo/deb/public.key | sudo apt-key add -

echo deb https://aquasecurity.github.io/trivy-repo/deb $(lsb\_release -sc) main | sudo tee -a /etc/apt/sources.list.d/trivy.list

sudo apt-get update

sudo apt-get install trivy

trivy repo –format json -o report.json [https://github.com/<Git-url](https://github.com/%3cGit-url)>

trivy repo –format table -o report.html [https://github.com/<Git-url](https://github.com/%3cGit-url)>

Format types:

1. table

2. Json

3. template ..ect

How we can scan the file system

1. clone the github Repo URL

- git clone <[https://github.com/......./ file.git](https://github.com/......./%20file.git)>

2. Scan the inside the folder command

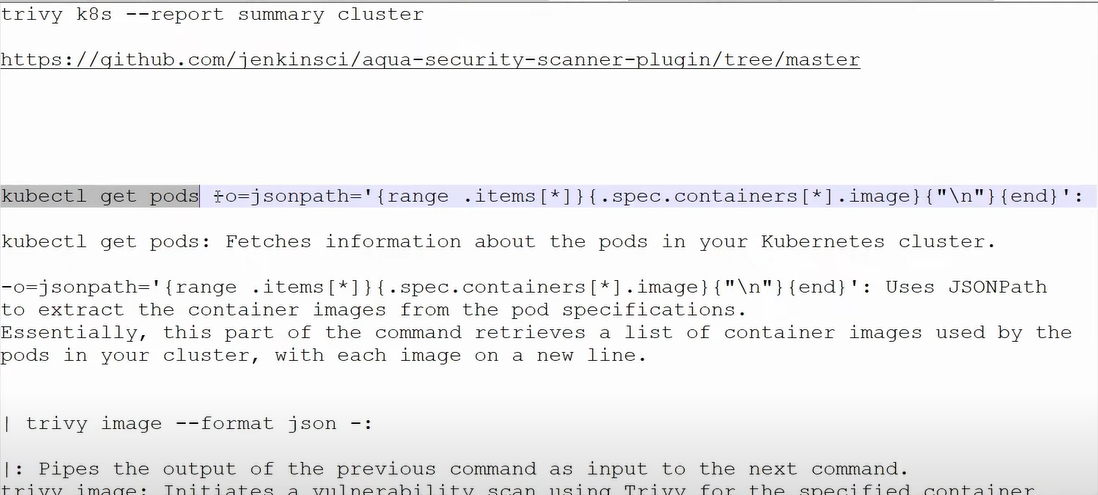
- trivy fs .

3. Scan the image.

- trivy image sonarqube

- trivy image --format table -o docker-report.txt sonarqube

**K8s Scan Commands:**



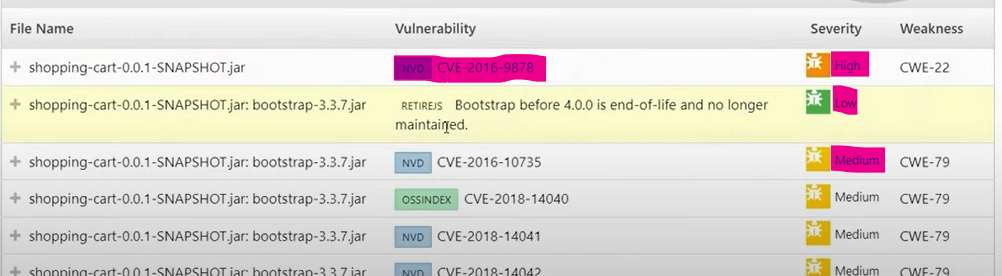
**OWASP Dependency Check:**

**1. What is Vulnerabilities.**

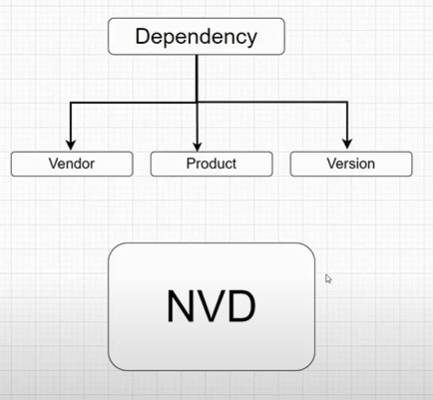
**2.what is CPE**

**3. What is CVE**

**4. What is CVSS**

****

**OS Dependency Check:**

****

**EX:** For the above process we have team as CPE(Common Platform Enumeration) using this division will be happening.A screenshot of a computer

Description automatically generated



**NVD ---> National Vulnerability database**

**--** This Data base it contains all the information about different vulnerabilities different kind of vulnerabilities that already found out.

**What is Dependency-Check?**

🡺 Dependency-Check is a Software Composition Analysis (SCA) tool that attempts to detect publicly disclosed vulnerabilities contained within a project’s dependencies. It does this by determining if there is a [Common Platform Enumeration](https://nvd.nist.gov/products/cpe) (CPE) identifier for a given dependency.

🡺 The core engine contains a series of analysers that inspect the project dependencies and identify the CPE for the given dependency. If a CPE is identified then it is cross referenced to the [NIST CVE database](https://nvd.nist.gov/) and any associated [Common Vulnerability and Exposure](https://cve.mitre.org/) (CVE) entries are listed in the report.

Dependency-Check’s core analysis engine can be used as: -

* an Ant Task
* a Command Line Tool
* Gradle Plugin
* Jenkins Plugin
* Maven Plugin
* SBT Plugin

**Why use it?**

🡺 Checking for vulnerable components, ‘[A06 Vulnerable and Outdated Components](https://owasp.org/Top10/A06_2021-Vulnerable_and_Outdated_Components/)’, is in the OWASP Top Ten and is one of the most straight-forward and effective security activities to implement. The Dependency-Check tool provides checks for vulnerable components that can be run from the command line.

🡺This is useful for development teams; the ability to check for vulnerable application components from the command line gives immediate feedback to the developer without having to wait for a pipeline to run.

🡺Dependency-Check also provides plugins to check for vulnerable components for [CI/CD pipelines](https://cheatsheetseries.owasp.org/cheatsheets/CI_CD_Security_Cheat_Sheet).

**How to use it?**

🡺 The OWASP Spotlight series provides an example of the risks involved in using out of date and vulnerable libraries, and how to use Dependency-Check: ‘Project 2 - [OWASP Dependency Check](https://youtu.be/YAXf3TaAYeA)’.

**Refer to the Dependency-Check**[**documentation**](https://jeremylong.github.io/DependencyCheck/)**to get started running from the command line:**

* ensure Java is installed, for example from [Eclipse Adoptium](https://adoptium.net/)
* download and unzip the latest Dependency-Check [release](https://github.com/jeremylong/DependencyCheck/releases)
* navigate to the Dependency-Check ‘bin’ directory and run, using threat Dragon as an example: ./dependency-check.sh --project "Threat Dragon" --scan ~/github/threat-dragon.
* open the html output file and act on the findings.

The command line is useful for immediate debugging development. Depending on what automation environment is in place a plugin can also be installed into a pipeline which can then generate the SCA reports.