

Композиционный анализ зависимостей проекта

--

Для начала работ вам понадобится:

- *docker*
- *git*

Подробная установка требуемых инструментов подробно рассмотрена на предыдущих уроках

--

Часть 1. Подготовка к композиционному сканированию

Создадим shell-скрипт для сборки образа *dependency-check* и запуска с его помощью сканера композиционного анализа:

```
$ nano dependency-check.sh
```



Добавим в файл следующий контент и сохраним изменения:

```
#!/bin/sh
```

```
DC_VERSION="latest"  
DC_DIRECTORY=$HOME/OWASP-Dependency-Check  
DC_PROJECT="dependency-check scan: $(pwd) "  
DATA_DIRECTORY="$DC_DIRECTORY/data"  
CACHE_DIRECTORY="$DC_DIRECTORY/data/cache"
```

```

if [ ! -d "$DATA_DIRECTORY" ]; then
    echo "Initially creating persistent directory:
$DATA_DIRECTORY"
    mkdir -p "$DATA_DIRECTORY"
fi
if [ ! -d "$CACHE_DIRECTORY" ]; then
    echo "Initially creating persistent directory:
$CACHE_DIRECTORY"
    mkdir -p "$CACHE_DIRECTORY"
fi

# Make sure we are using the latest version
docker pull owasp/dependency-check:$DC_VERSION

docker run --rm \
    -e user=$USER \
    -u $(id -u ${USER}):$(id -g ${USER}) \
    --volume $(pwd):/src:z \
    --volume "$DATA_DIRECTORY":/usr/share/dependency-
check/data:z \
    --volume $(pwd)/odc-reports:/report:z \
    owasp/dependency-check:$DC_VERSION \
    --scan /src \
    --format "ALL" \
    --project "$DC_PROJECT" \
    --out /report
    # Use suppression like this: (where /src == $pwd)
    # --suppression "/src/security/dependency-check-
suppression.xml"

```

В качестве проверяемого проекта будем использовать сам проект owasp dependency-check, для этого склонируем репозиторий:

```

$ git clone
https://github.com/jeremylong/DependencyCheck.git

```

```

filipp@filipp-notebook:~/Desktop$ git clone https://github.com/jeremylong/DependencyCheck.git
Cloning into 'DependencyCheck'...
remote: Enumerating objects: 171258, done.
remote: Counting objects: 100% (196/196), done.
remote: Compressing objects: 100% (109/109), done.
remote: Total 171258 (delta 89), reused 145 (delta 69), pack-reused 171062
Receiving objects: 100% (171258/171258), 203.28 MiB | 2.86 MiB/s, done.
Resolving deltas: 100% (116173/116173), done.

```

Поместим shell-скрипт dependency-check.sh в директорию с проектом:

```
$ cp dependency-check.sh DependencyCheck/
```

```
filipp@filipp-notebook:~/Desktop$ cp dependency-check.sh DependencyCheck/  
filipp@filipp-notebook:~/Desktop$
```

Выдадим необходимые права скрипту для запуска:

```
$ chmod +x dependency-check.sh
```

```
filipp@filipp-notebook:~/Desktop/DependencyCheck$ chmod +x dependency-check.sh  
filipp@filipp-notebook:~/Desktop/DependencyCheck$
```

--

Часть 2. Проведение композиционного сканирования зависимостей проекта

Запустим подготовленный скрипт композиционного сканирования

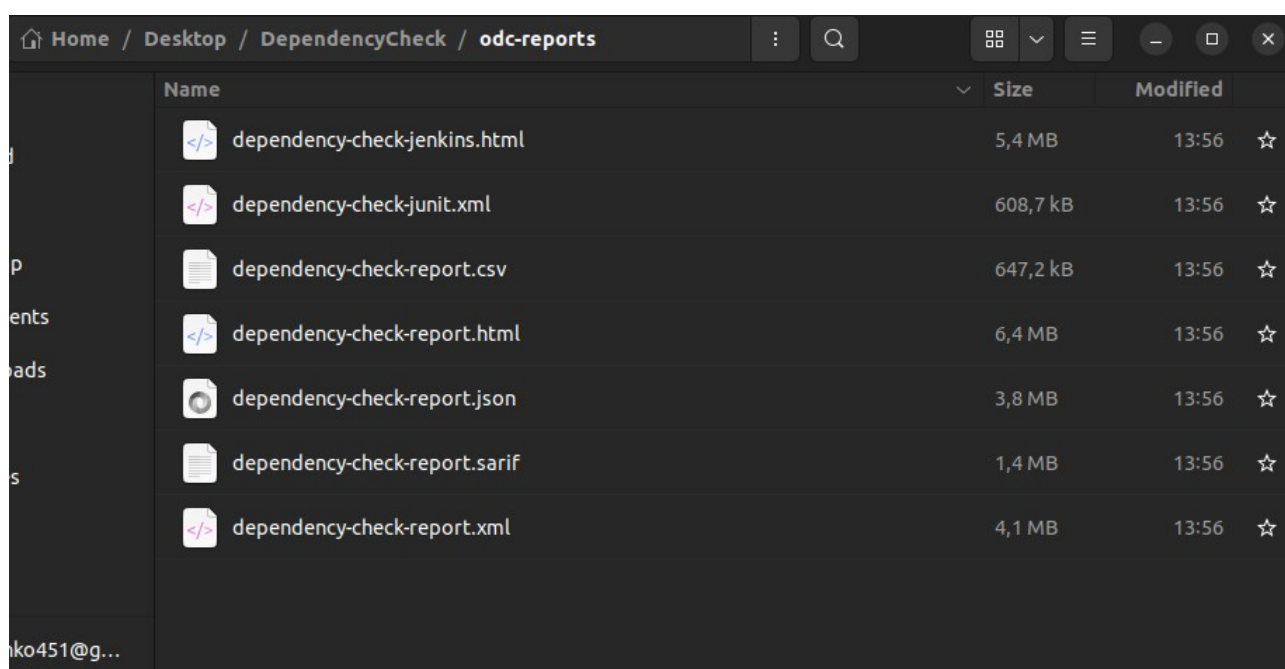
```
$ sudo ./dependency-check.sh
```








```
filipp@filipp-notebook:~/Desktop/DependencyCheck$ sudo ./dependency-check.sh  
latest: Pulling from owasp/dependency-check  
Digest: sha256:3b48f8bfadc3c689a80b6cefe3efd149d124071b34cfb22e833bb633f7fc137d  
Status: Image is up to date for owasp/dependency-check:latest  
docker.io/owasp/dependency-check:latest
```

В ходе анализа может потребоваться обновление баз вирусных сигнатур. По результатам сканирования будет сформировано несколько видов отчетов:

```
[INFO] Finished Sonatype OSS Index Analyzer (11 seconds)
[INFO] Finished Vulnerability Suppression Analyzer (0 seconds)
[INFO] Finished Known Exploited Vulnerability Analyzer (0 seconds)
[INFO] Finished Dependency Bundling Analyzer (0 seconds)
[INFO] Finished Unused Suppression Rule Analyzer (0 seconds)
[INFO] Analysis Complete (21 seconds)
[INFO] Writing report to: /report/dependency-check-report.xml
[INFO] Writing report to: /report/dependency-check-report.html
[INFO] Writing report to: /report/dependency-check-report.json
[INFO] Writing report to: /report/dependency-check-report.csv
[INFO] Writing report to: /report/dependency-check-report.sarif
[INFO] Writing report to: /report/dependency-check-jenkins.html
[INFO] Writing report to: /report/dependency-check-junit.xml
```

Перейдем в сформированный dependency-check каталог odc-reports:



Name	Size	Modified
 dependency-check-jenkins.html	5,4 MB	13:56 ☆
 dependency-check-junit.xml	608,7 kB	13:56 ☆
 dependency-check-report.csv	647,2 kB	13:56 ☆
 dependency-check-report.html	6,4 MB	13:56 ☆
 dependency-check-report.json	3,8 MB	13:56 ☆
 dependency-check-report.sarif	1,4 MB	13:56 ☆
 dependency-check-report.xml	4,1 MB	13:56 ☆

Откроем отчет в формате html:

Summary

Display: [Showing Vulnerable Dependencies \(click to show all\)](#)

Dependency	Vulnerability IDs	Package	Highest Severity ↓	CVE Count	Confidence
sinatra/Gemfile.lock:rack	cpe:2.3:a:rack_project:rack:1.5.2:*:*:*:*:*	pkg:gem/rack@1.5.2	CRITICAL	10	High
sinatra/Gemfile.lock:rack-protection	cpe:2.3:a:rack_project:rack:1.5.2:*:*:*:*:*	pkg:gem/rack-protection@1.5.2	CRITICAL	10	High
rails-4.1.15/Gemfile.lock:rack	cpe:2.3:a:rack_project:rack:1.5.5:*:*:*:*:*	pkg:gem/rack@1.5.5	CRITICAL	9	High
uber-1.0-SNAPSHOT.jar (shaded: org.eclipse.jetty:jetty-util:7.6.0.RC4)	cpe:2.3:a:eclipse:jetty:7.6.0.rc4:*:*:*:*:* cpe:2.3:a:jetty:jetty:7.6.0.rc4:*:*:*:*:*	pkg:maven/org.eclipse.jetty/jetty-util@7.6.0.RC4	CRITICAL	11	Highest
uber-1.0-SNAPSHOT.jar (shaded: org.eclipse.jetty:jetty-servlet:7.6.0.RC4)	cpe:2.3:a:eclipse:jetty:7.6.0.rc4:*:*:*:*:* cpe:2.3:a:jetty:jetty:7.6.0.rc4:*:*:*:*:*	pkg:maven/org.eclipse.jetty/jetty-servlet@7.6.0.RC4	CRITICAL	11	Highest
uber-1.0-SNAPSHOT.jar (shaded: org.eclipse.jetty:jetty-server:7.6.0.RC4)	cpe:2.3:a:eclipse:jetty:7.6.0.rc4:*:*:*:*:* cpe:2.3:a:jetty:jetty:7.6.0.rc4:*:*:*:*:*	pkg:maven/org.eclipse.jetty/jetty-server@7.6.0.RC4	CRITICAL	11	Highest
uber-1.0-SNAPSHOT.jar (shaded: org.eclipse.jetty:jetty-security:7.6.0.RC4)	cpe:2.3:a:eclipse:jetty:7.6.0.rc4:*:*:*:*:* cpe:2.3:a:jetty:jetty:7.6.0.rc4:*:*:*:*:*	pkg:maven/org.eclipse.jetty/jetty-security@7.6.0.RC4	CRITICAL	11	Highest
uber-1.0-SNAPSHOT.jar (shaded: org.eclipse.jetty:jetty-io:7.6.0.RC4)	cpe:2.3:a:eclipse:jetty:7.6.0.rc4:*:*:*:*:* cpe:2.3:a:jetty:jetty:7.6.0.rc4:*:*:*:*:*	pkg:maven/org.eclipse.jetty/jetty-io@7.6.0.RC4	CRITICAL	11	Highest
uber-1.0-SNAPSHOT.jar (shaded: org.eclipse.jetty:jetty-http:7.6.0.RC4)	cpe:2.3:a:eclipse:jetty:7.6.0.rc4:*:*:*:*:* cpe:2.3:a:jetty:jetty:7.6.0.rc4:*:*:*:*:*	pkg:maven/org.eclipse.jetty/jetty-http@7.6.0.RC4	CRITICAL	11	Highest
uber-1.0-SNAPSHOT.jar (shaded: org.eclipse.jetty:jetty-continuation:7.6.0.RC4)	cpe:2.3:a:eclipse:jetty:7.6.0.rc4:*:*:*:*:* cpe:2.3:a:jetty:jetty:7.6.0.rc4:*:*:*:*:*	pkg:maven/org.eclipse.jetty/jetty-continuation@7.6.0.RC4	CRITICAL	11	Highest

На основе полученных данных в отчете можно сформулировать результат композиционного сканирования зависимостей проекта

--