Сканирование образа docker тремя различными сканерами

--

Для начала работ вам понадобится:

- *docker*

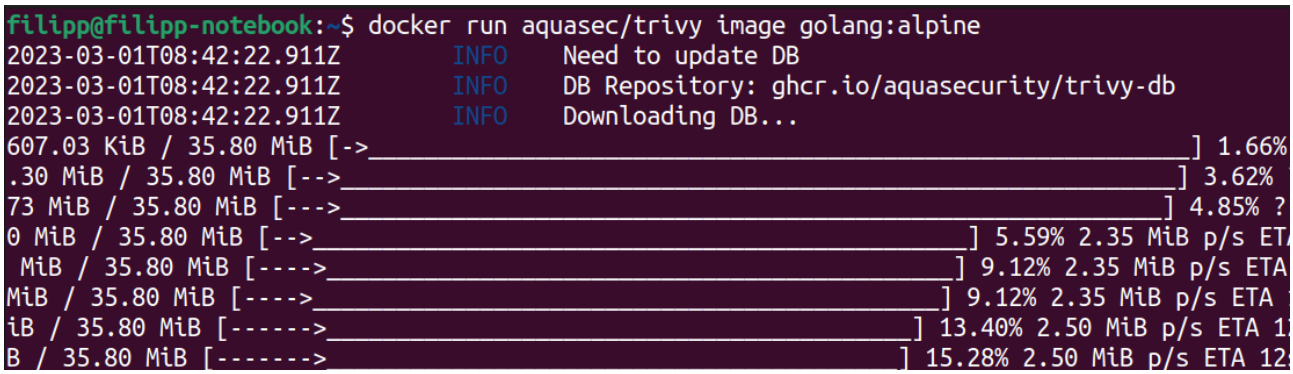
Подробная установка docker рассматривалась в методических материалах к предыдущим урокам

--

Часть 1. Установка и использование сканера trivy

Одновременно производим запуск сканера trivy и проводим проверку образа `golang:alpine`:

```
$ docker run aquasec/trivy image golang:alpine
```



Дожидаемся инициализации баз данных вирусных сигнатур и завершения сканирования:

```
golang:alpine (alpine 3.17.2)
=============================
Total: 0 (UNKNOWN: 0, LOW: 0, MEDIUM: 0, HIGH: 0, CRITICAL: 0)
```

Запомним результаты сканирования инструментом trivy для дальнейшего анализа

--

Часть 2. Установка и использование сканера clair

Скачаем с github проект clair:

```
$ curl -L
https://github.com/arminc/clair-scanner/releases/download
/v12/clair-scanner_linux_amd64 -o /usr/bin/clair-scanner
```

```
filipp@filipp-notebook:~/Desktop$ curl -L https://github.com/arminc/clair-scanner/releases/download/v
12/clair-scanner_linux_amd64 -o /usr/bin/clair-scanner
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
100 9631k  100 9631k    0     0  2358k      0  0:00:04  0:00:04 --:--:-- 2962k
```

Добавим прав сканеру clair:

```
$ chmod +x /usr/bin/clair-scanner
```

```
filipp@filipp-notebook:~/Desktop$ chmod +x /usr/bin/clair-scanner
filipp@filipp-notebook:~/Desktop$ ll /usr/bin/clair-scanner
-rwxrwxr-x 1 filipp filipp 9862522 мар  1 11:47 /usr/bin/clair-scanner*
```

Запускаем в контейнере базу данных clair:

```
$ docker run -p 5432:5432 -d --name db arminc/clair-
db:latest
```

```
filipp@filipp-notebook:~/Desktop$ docker run -p 5432:5432 -d --name db arminc/clair-db:latest
98aa50e0bd9e366a088cbf6c95a5d886441049b89babf0156601e7b00ae16df3
filipp@filipp-notebook:~/Desktop$ docker ps
CONTAINER ID   IMAGE                  COMMAND                  CREATED         STATUS         PORTS
                                  NAMES
98aa50e0bd9e   arminc/clair-db:latest   "docker-entrypoint.s…"   3 seconds ago   Up 3 seconds   0.0.0
.0:5432->5432/tcp, :::5432->5432/tcp   db
```

Запускаем в контейнере локальный сканер clair:

```
$ docker run -p 6060:6060 --link db:postgres -d --name
clair arminc/clair-local-scan:latest
```

```
filipp@filipp-notebook:~/Desktop$ docker run -p 6060:6060 --link db:postgres -d --name clair arminc/c
lair-local-scan:latest
98a9deabe850f030ff4ae650d66169f6d029633f7f5696e8bb7d67ec0265d536
filipp@filipp-notebook:~/Desktop$ docker ps
CONTAINER ID   IMAGE                             COMMAND                  CREATED              STATUS
             PORTS                                                NAMES
98a9deabe850   arminc/clair-local-scan:latest    "/clair -config=/con…"   4 seconds ago        Up 3 se
conds        0.0.0.0:6060->6060/tcp, :::6060->6060/tcp, 6061/tcp   clair
98aa50e0bd9e   arminc/clair-db:latest            "docker-entrypoint.s…"   About a minute ago   Up Abou
t a minute   0.0.0.0:5432->5432/tcp, :::5432->5432/tcp             db
```

Запустим сканирование образа `golang:alpine` с помощью clair:

```
$ clair-scanner -r php-report.json --ip 172.17.0.1
golang:alpine
```

```
filipp@filipp-notebook:~/Desktop$ clair-scanner -r php-report.json --ip 172.17.0.1 golang:alpine
2023/03/01 12:23:40 [INFO] ▶ Start clair-scanner
2023/03/01 12:23:47 [INFO] ▶ Server listening on port 9279
2023/03/01 12:23:47 [INFO] ▶ Analyzing 921ba1c4aa74416c0e837da593035fa69fd47d6897a87802d105d5f580a0f5
c8
2023/03/01 12:23:48 [INFO] ▶ Analyzing a07d6622aabd49c4b0e388c961f89ab361e279205b4da245ba96123f4338df
1d
2023/03/01 12:23:48 [INFO] ▶ Analyzing 8790190c883d7c2a4cc1c915f1e5e3da52baeb5d340d87f8e01e8176322677
21
2023/03/01 12:23:48 [INFO] ▶ Analyzing 46c429d274f5d81071d97d936ce931249587e3eda6541564f81cb012eb9616
c6
2023/03/01 12:23:48 [INFO] ▶ Image [golang:alpine] contains NO unapproved vulnerabilities
```

Запомним результаты сканирования инструментом trivy для дальнейшего анализа

--

Часть 3. Сканирование с помощью snyk

Выполним login в dockerhub:

```
$ docker login
```

```
filipp@filipp-notebook:~/Desktop$ docker login
Login with your Docker ID to push and pull images from Docker Hub. If you don't have a
d over to https://hub.docker.com to create one.
Username: ignatenko451
Password:
```

Выполним сканирование образа `golang:alpine` с помощью snyk:

```
$ docker scan golang:alpine
```

```
filipp@filipp-notebook:~/Desktop$ docker scan golang:alpine

Testing golang:alpine...
```

Дождемся завершения сканирования и запомним результаты:

```
Package manager:   apk
Project name:      docker-image|golang
Docker image:      golang:alpine
Platform:          linux/amd64
Base image:        golang:1.19.3-alpine

Tested 15 dependencies for known vulnerabilities, found 4 vulnerabilities.
```

--

Часть 4. Сравнение результатов анализа

По результатам сканирования, trivy и clair уязвимсоетй в образе не нашли, snyk нашел 4 потенциальных уязвимости

--

Часть 5. Дополнительное сканирование образа mariadb

Запустим новое сканирование для образа `mariadb:latest` всеми тремя инструментами:

```
$ docker run aquasec/trivy image mariadb:latest
```

```
mariadb:latest (ubuntu 22.04)
==============================
Total: 37 (UNKNOWN: 0, LOW: 16, MEDIUM: 19, HIGH: 2, CRITICAL: 0)
```

```
$ clair-scanner -r php-report.json --ip 172.17.0.1
mariadb:latest
```

```
filipp@filipp-notebook:~/Desktop$ clair-scanner -r php-report.json --ip 172.17.0.1 mariadb:latest
2023/03/01 12:43:54 [INFO] ▶ Start clair-scanner
2023/03/01 12:43:57 [INFO] ▶ Server listening on port 9279
2023/03/01 12:43:57 [INFO] ▶ Analyzing 18d871242eb6f10177f837472e5d5d6182e4e7ec2c7ef60bb58fd39be3cf4c2a
2023/03/01 12:43:59 [INFO] ▶ Analyzing b63bfe2b710c98160832df804b609003a21cb02241700df291bd4cde8272e818
2023/03/01 12:43:59 [INFO] ▶ Analyzing 2ab410470aa86135bccaa4be7e49375133dd107a4f5b840341d4c0a26c1f5073
2023/03/01 12:43:59 [INFO] ▶ Analyzing 99b5685d37e38f9e4f36e7c4bb1f34826df29c764230139227262d476b9d9291
2023/03/01 12:43:59 [INFO] ▶ Analyzing 660c482c0dfb0dbf6fe73bc5e50a77405ba4fa9e55a9ed10fcd0c0a4fdbe116e
2023/03/01 12:43:59 [INFO] ▶ Analyzing 72b9a41124bdbe6c59a42b57b70f46a892bf1e34c43a9401ca27838bdb99b54c
2023/03/01 12:43:59 [INFO] ▶ Analyzing c00ddc2f503e8f89d8de086455d330e695647a8ee28d28dafa0f1076315557a1
2023/03/01 12:43:59 [INFO] ▶ Analyzing 2ba102a5c7c28e0098c82e7f59b454aaeec77515ecc8bc1159128353736e0505
2023/03/01 12:43:59 [INFO] ▶ Image [mariadb:latest] contains NO unapproved vulnerabilities
```

```
$ docker scan mariadb:latest
```

```
Package manager:   deb
Project name:      docker-image|mariadb
Docker image:      mariadb:latest
Platform:          linux/amd64

Tested 154 dependencies for known vulnerabilities, found 21 vulnerabilities.
```

По результатам сканирования: trivy обнаружено 37 уязвимостей, clair не обнаружено, snyk обнаружено 21 потенциальная уязвимость

--