

Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное образовательное учреждение
высшего образования «Национальный исследовательский университет
ИТМО»

Факультет инфокоммуникационных технологий

Основы кибербезопасности

Практическая работа №3

Выполнил:

студент группы К34211

Фисенко Максим Вячеславович

Проверил:

преподаватель практики, КТН

Назаров Михаил Сергеевич

Санкт-Петербург

2024

Оглавление

Введение	3
Содержание отчета.....	4
Задание 1. Проверка корректности работы Docker	4
Задание 2. Создание лаборатории для тестирования и поиска уязвимостей	5
Задание 3. Работа со сканером уязвимостей OpenVAS	7
Вывод по работе.....	11

Введение

В данной практической работе было необходимо изучить типовой алгоритм работы с инструментами обнаружения уязвимостей информационных систем. В ходе практической работы были приобретены практические навыки по использованию сканера уязвимостей, а также по идентификации уязвимостей информационной системы.

Для выполнения данной практической работы использовался персональный компьютер на операционной системе *Windows* с подключенной через *WSL* дистрибутивом *Linux Ubuntu*.

Содержание отчета

Задание 1. Проверка корректности работы Docker

Первым делом для выполнения данной практической работы необходимо было проверить работоспособность *Docker* на устройстве. Для этого в терминале была предпринята попытка запустить простой образ *hello-world*. Как видно на рисунке 1 ниже, контейнер с образом был успешно запущен, а это значит, что *Docker* на устройстве работает.

```
maksim@pc:~$ docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
c1ec31eb5944: Pull complete
Digest: sha256:305243c734571da2d100c8c8b3c3167a098cab6049c9a5b066b6021a60fcb966
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (amd64)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
   to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/
```

Рисунок 1 - Запуск контейнера hello-world

Так как далее при выполнении лабораторной работы необходимо будет поднимать свой веб-сервер, необходимо было проверить, нет ли с этим проблем на данном устройстве. Для этого на порту 80 был запущен контейнер *webserver nginx* с помощью ввода в терминал команды *docker run --detach --publish=80:80 --name=webserver nginx*. Контейнер был успешно запущен, и при вводе в адресную строку *http://localhost* успешно отображалась приветственная страница (рисунок 2).



Рисунок 2 - Приветственная страница *webserver nginx*

Далее в терминал была введена команда *docker container ls*, которая показывает информацию о всех запущенных контейнерах, а в данном случае, о контейнере *webserver nginx* (рисунок 3).

```
maksim@pc:~$ docker container ls
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
01fcb81e776a	nginx	"/docker-entrypoint..."	About a minute ago	Up About a minute	0.0.0.0:80->80/tcp	webserver

Рисунок 3 - Информация о запущенных контейнерах

Задание 2. Создание лаборатории для тестирования и поиска уязвимостей

После проверки работоспособности *Docker* нужно было скачать на устройство образы, которые будут необходимы для выполнения практической работы. Это было сделано с помощью команды *docker pull*. На рисунке 4 с помощью команды *docker images* были выведены все образы, скачанные на устройство, из которых *metasploitable2*, *kali-rolling* и *openvas* – те самые образы, которые необходимы для выполнения практической работы.

```
maksim@pc:~$ docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
nginx	latest	1ee494ebb83f	3 days ago	192MB
kalilinux/kali-rolling	latest	4df3f4beda4d	6 days ago	129MB
mysql	latest	10db11fef9ce	6 weeks ago	602MB
mysql	8.0	9f4b39935f20	6 weeks ago	590MB
postgres	16.3	cff6b68a194a	6 months ago	432MB
sonarqube	10.5.1-community	b728f044f72f	7 months ago	787MB
hello-world	latest	d2c94e258dcb	19 months ago	13.3kB
mikesplain/openvas	latest	889967897c49	5 years ago	6.39GB
tleemcjr/metasploitable2	latest	db90cb788ea1	6 years ago	1.51GB

Рисунок 4 - Скачанные на устройство образы

Далее необходимо было запустить контейнеры с нужными образами. Для этого первым делом с помощью команды *docker network create pentest* была создана сеть *pentest*, в которой и будут работать контейнеры. Затем с помощью *docker run* были запущены и сами контейнеры с именами *metasploitable2* и *kalibox*. При этом данные контейнеры были запущены в интерактивном режиме в двух разных терминалах.

После запуска контейнеров необходимо было убедиться в том, что контейнеры видят друг друга в сети. Для этого сначала в *kalibox* была использована утилита *ifconfig*, благодаря которой мы узнали *ip*-адрес данного контейнера (рисунок 5). Стоит отметить, что для ее использования предварительно был установлен пакет *net-tools*.

```
(root@attacker)-[/]
# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 172.18.0.3 netmask 255.255.0.0 broadcast 172.18.255.255
    ether 02:42:ac:12:00:03 txqueuelen 0 (Ethernet)
    RX packets 15169 bytes 21917599 (20.9 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 4965 bytes 328946 (321.2 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Рисунок 5 - Утилита *ifconfig*

В свою очередь, в контейнере *metasploitable2* была использована утилита *ping* на адрес, который был узнан на предыдущем шаге. Как видно из рисунка 6 ниже, *ICMP*-пакеты успешно пересылаются между двумя контейнерами, а это значит, что они успешно видят друг друга в сети.

```
root@victim:/# ping 172.18.0.3
PING 172.18.0.3 (172.18.0.3) 56(84) bytes of data.
64 bytes from 172.18.0.3: icmp_seq=1 ttl=64 time=0.052 ms
64 bytes from 172.18.0.3: icmp_seq=2 ttl=64 time=0.030 ms
64 bytes from 172.18.0.3: icmp_seq=3 ttl=64 time=0.030 ms
64 bytes from 172.18.0.3: icmp_seq=4 ttl=64 time=0.039 ms
64 bytes from 172.18.0.3: icmp_seq=5 ttl=64 time=0.031 ms
^C
--- 172.18.0.3 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 6862ms
rtt min/avg/max/mdev = 0.030/0.036/0.052/0.010 ms
```

Рисунок 6 - Утилита *ping*

Затем в контейнере *metasploitable2* была создана новая учетная запись *maksim* (рисунок 7).

```

root@victim:/# useradd maksim
root@victim:/# passwd maksim
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
root@victim:/# usermod -aG sudo maksim

```

Рисунок 7 – Создание новой учетной записи

Наконец, был запущен контейнер с *openvas*. Контейнер был запущен на порту 443 с помощью следующей команды:

```
docker run --network=pentest -d -p 443:443 --name openvas mikesplain/openvas
```

Таким образом, после запуска данного контейнера на устройстве работало одновременно 3 контейнера, что было проверено опять же с помощью команды *docker container ls* (рисунок 8).

```

maksim@pc:~$ docker container ls
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS                               NAMES
46102722cc45   mikesplain/openvas                  "/bin/sh -c /start"     2 minutes ago Up 2 minutes  0.0.0.0:443->443/tcp, 9390/tcp      openvas
9cef9722c3a6   kalilinux/kali-rolling              "bash"                  9 minutes ago Up 9 minutes                               kalibox
b64f52f922cf   tleemcjr/metasploitable2           "sh -c '/bin/service..." 11 minutes ago Up 11 minutes                               metasploitable2

```

Рисунок 8 – Три работающих контейнера

Задание 3. Работа со сканером уязвимостей OpenVAS

Так как контейнер с *openvas* был успешно запущен, можно было переходить уже непосредственно к работе со сканером уязвимости. Для этого необходимо было открыть страницу *https://localhost*. Крайне важно, что необходимо было открыть именно *httpS://localhost*. Мною, к сожалению, это изначально не было сделано, из-за чего довольно много времени было потрачено на поиск и решение проблемы. В итоге страница была открыта, и стартовая страница *openvas* успешно отображалась в браузере (рисунок 9).

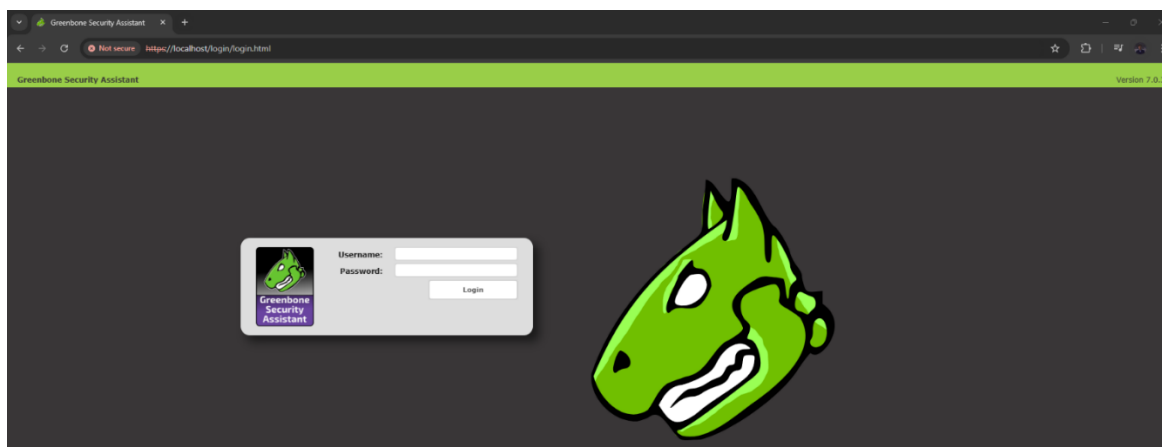


Рисунок 9 – Стартовая страница openvas

В форме было введено *admin* в качестве логина и пароля, после чего была открыта главная страница *openvas*. Первым делом на данной странице было необходимо завести учетную запись для проведения локальных проверок, что и было сделано – была создана учетная запись с именем *maksim* (рисунок 10).

Рисунок 10 – Создание учетной записи

Затем было необходимо задать цель сканирования. В качестве цели необходимо было выбрать контейнер *metasploit2*, *ip*-адрес которого был обнаружен с помощью всё той же утилиты *ping*. В окне были заполнены все необходимые данные о цели сканирования, включающие в себя и данный *ip*-адрес (рисунок 11).

Рисунок 11 – Создание цели сканирования

После того, как была задана цель сканирования, в *openvas* необходимо было создать задачу. В окне *New Task* было указано название задачи, а в поле *Scan Targets* была добавлена созданная на предыдущем шаге цель (рисунок 12).

Рисунок 12 – Создание задачи

Наконец, все приготовления были закончены, и процесс сканирования был начат при нажатии на кнопку. Для удобства сверху была выбрана опция автообновления страницы каждые 5 минут. В результате, где-то в течение часа

процесс сканирования был завершен, и отчет об этом появился на странице (рисунок 13).

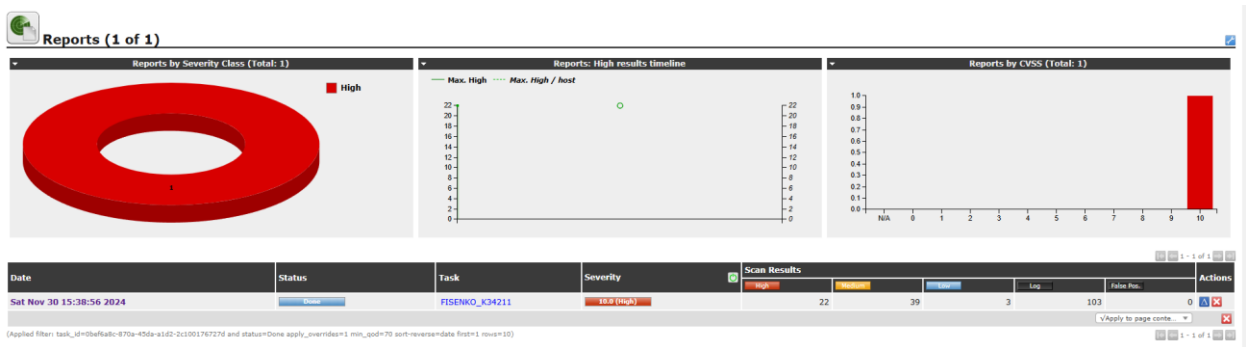


Рисунок 13 – Отчет о сканировании

Как видно на рисунке 13, представленном выше, сканер нашел достаточно большое количество угроз, из которых 22 угрозы имеют наивысший статус опасности. При более подробном рассмотрении отчета о сканировании можно посмотреть на все найденные угрозы (рисунок 14).

Report: Results (64 of 405)

ID: e5c14d11-7962-4f00-b083-4346b8aeb93

Modified: Sat Nov 30 16:05:12 2024

Created: Sat Nov 30 15:39:03 2024

Owner: admin

</

Рисунок 14 – Список найденных угроз по уменьшению опасности

Вывод по работе

В результате выполнения данной лабораторной работы на персональном компьютере была развернута среда *OpenVAS*, которая просканировала контейнер с сервером *metasploit2* и обнаружила в нем уязвимости, показав каждую из них и отсортировав их список по степени опасности. Помимо *openvas*, в ходе выполнения работы был активно задействован *Docker*, а также утилиты *ping* и *ifconfig*. По результатам выполнения работы можно с уверенностью сказать, что цель практической работы была достигнута.