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

**Выполнила**:

студент группы К34211

Фисенко Максим Вячеславович

**Проверил**:

преподаватель практики, КТН

Назаров Михаил Сергеевич

Санкт-Петербург

2024

**Оглавление**

**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 и подключенной через WSL операционной системой Ubuntu.

**Содержание отчета**

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

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

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).

Изображение выглядит как текст, программное обеспечение, снимок экрана, веб-страница

Автоматически созданное описание

Рисунок 2 - Приветственная страница webserver nginx

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

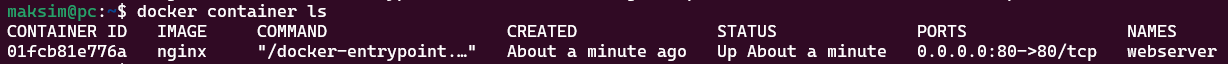


Рисунок 3 - Информация о запущенных контейнерах

После проверки работоспособности Docker было необходимо 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 былы выведены все образы, скачанные на устройство, из которых metasploitable2,kali-rolling и openvas – те самые образы, которые необходимы для выполнения практической работы.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

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 конейтеры 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.

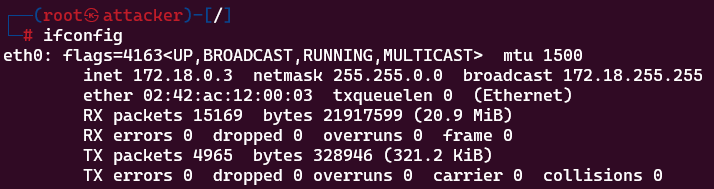


Рисунок 5 - Утилита ifconfig

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.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 6 - Утилита ping

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

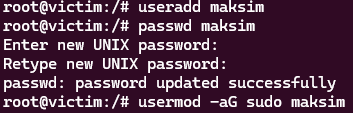


Рисунок 7 – Создание новой учетной записи

Наконец, был запущен контейнер с openvas. Контейнер был запущен на порту 443:

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).

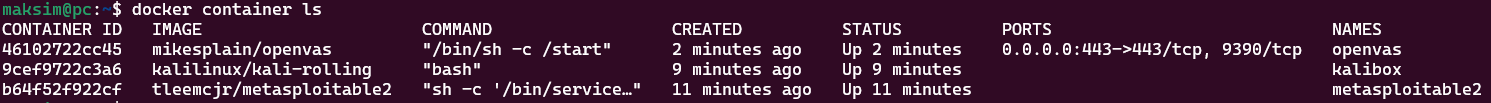


Рисунок 8 – 3 работающих контейнера

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 через httpS. 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 был введен admin в качестве логина и паролю, 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).

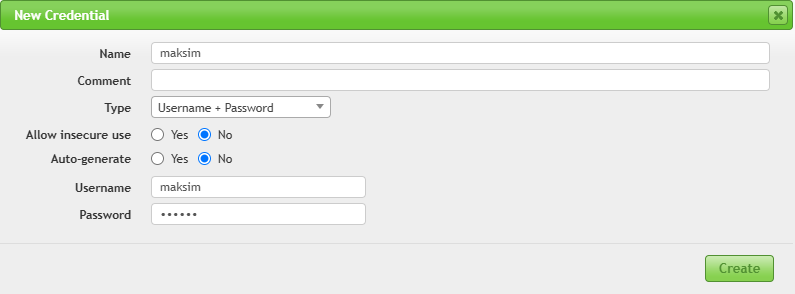


Рисунок 10 – Создание учетной записи

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).

Изображение выглядит как текст, программное обеспечение, веб-страница, Значок на компьютере

Автоматически созданное описание

Рисунок 11 – Создание цели сканирования

**Вывод по работе**

**Литература (ссылки)**