

**Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Факультет прикладной информатики (ФПИИ)

Лабораторная работа №8

по дисциплине: Основы кибербезопасности

Автор: доцент практики, кандидат технических наук

Кравчук Алексей Владимирович

Санкт-Петербург
2025

Тема занятия: Обнаружение и анализ инфраструктурных уязвимостей

Цель работы:

- Изучить типовой алгоритм работы с инструментами обнаружения уязвимостей информационных систем.
- Приобрести практические навыки по использованию сканера инфраструктурных уязвимостей.
- Научиться идентифицировать уязвимости информационной системы.

Краткие теоретические сведения

Материалы для самостоятельного изучения:

- <https://techcave.ru/posts/116-ustanovka-docker-na-mac-os.html>
- <https://www.digitalocean.com/community/tutorials/how-to-install-and-use-docker-on-ubuntu-20-04-ru>
- <https://propcwin.ru/kak-ustanovit-docker-na-windows-10.html>
- [OpenVAS. Безопасность системы! Проверка на уязвимость](#)
- [Видео установка Metasploitable2](#)

ссылки для скачивания Docker:

- [macos](#);
- [ubuntu](#);
- [windows](#).

Прочие дополнительные ссылки для скачивания:

- [VirtualBox](#)
- [Kali](#)
- [Metasploitable](#)

Практическая часть

1. Установка Docker

Добавьте официальный репозиторий Docker:

```
sudo apt update
sudo apt install -y ca-certificates curl gnupg
sudo install -m 0755 -d /etc/apt/keyrings
```

Добавьте GPG-ключ:

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | \
sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg
```

Добавьте сам репозиторий:

```
echo \
"deb [arch=$(dpkg --print-architecture) \
```

```
signed-by=/etc/apt/keyrings/docker.gpg] \
https://download.docker.com/linux/ubuntu \
$(lsb_release -cs) stable" | \
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

Обновляем кеш доступных пакетов и устанавливаем необходимые пакеты:

```
sudo apt update
sudo apt install -y docker.io docker-compose-plugin
```

Ранее можно было просто установить отдельную утилиту **docker-compose**. Теперь её использование не рекомендуется (deprecated). Поэтому устанавливаем плагин **docker-compose-plugin**.

Добавляем докер в автозагрузку:

```
sudo systemctl enable --now docker
```

Создаем рабочий каталог и переходим в него:

```
mkdir -p ~/lab8
cd ~/lab8
```

Разрешаем запуск docker без sudo:

```
sudo usermod -aG docker $USER
```

Выполняем проверку:

```
docker run --rm hello-world
```

```
user@lab8:~$ docker run --rm hello-world
Hello from Docker!
This message shows that your installation appears to be working correctly.
```

```
docker --version
```

```
user@lab8:~$ docker --version
Docker version 28.2.2, build 28.2.2-0ubuntu1~24.04.1
```

```
docker compose version
```

```
user@lab8:~$ docker compose version
Docker Compose version v2.40.3
```

Для развертывания контейнеров, удобнее использовать один файл `docker-compose.yml` и пару команд для останова и запуска:

```
docker compose down
docker compose up -d
```

2. Подготовка стенда

2.1. Скачать образы **metasploitable2** и **kali-rolling** для работы в **docker**

metasploitable 2 – уязвимый сервер (виртуальная машина).

В Metasploitable2 предусмотрено множество уязвимостей как на уровне операционной системы, так и на уровне сети и веб-приложений.

kali-rolling – контейнер (виртуальная машина) с инструментами используемыми для тестирования на проникновения

Введите команды на скачивание необходимых образов:

```
docker pull tleemcjr/metasploitable2
```

```
docker pull kalilinux/kali-rolling
```

Убедитесь, что образа скачались:

```
docker images
```

```
user@lab8:~/lab8_easy$ docker images
REPOSITORY          TAG         IMAGE ID      CREATED       SIZE
kalilinux/kali-rolling latest      b86fd25ecd19 3 days ago   124MB
tleemcjr/metasploitable2 latest      db90cb788ea1 7 years ago  1.51GB
```

2.2. Создать выделенную сеть **pentest** в **docker**

Создание сети *pentest* только для контейнеров выполняется путем ввода следующей команды

```
docker network create pentest
```

Выполните команду:

```
docker network ls
```

И убедитесь, что сеть создалась:

```
user@lab8:~/lab8_easy$ docker network ls
NETWORK ID          NAME        DRIVER       SCOPE
28f61d2fc85c       bridge     bridge      local
c4e4722223a5       host       host        local
cad187040e19       none       null        local
ebe294129eca       pentest    bridge      local
```

2.3. Создать 2 контейнера в выделенной сети **pentest**

Открыть два терминала.

В первом терминале выполнить следующую команду:

```
sudo docker run --network=pentest \  
    -h victim \  
    -it \  
    --rm \  
    --name metasploitable2 \  
    tleemcjr/metasploitable2
```

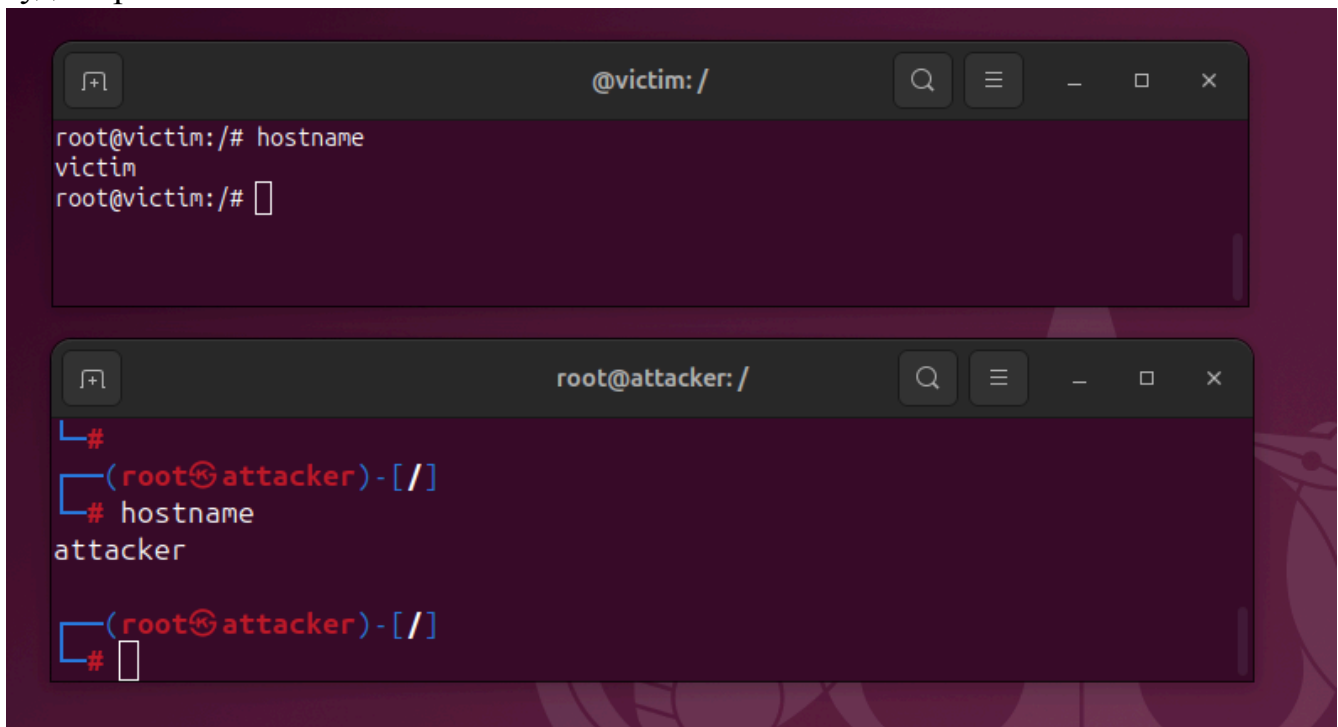
В результате docker создаст на основе образа *tleemcjr/metasploitable2* контейнер с именем *metasploitable2*, подключит его к сети *pentest* и задаст hostname внутри самой операционной системы контейнера равным *victim*.

Поскольку указана опция *-it* вы сможете попасть в интерактивную сессию контейнера (обычно shell). Когда вы выйдете из контейнера (Ctrl-D или exit), контейнер автоматически удалится (*--rm*).

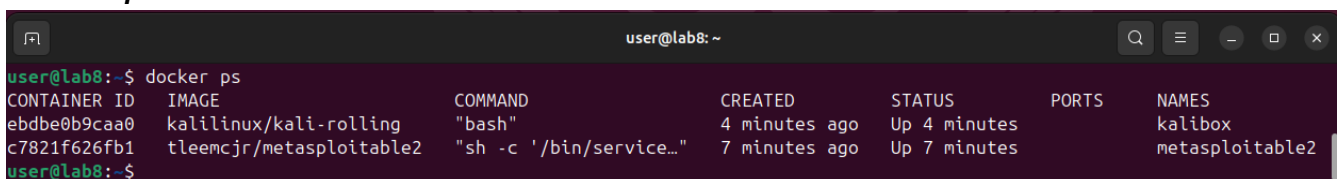
Во втором терминале выполните следующую команду:

```
sudo docker run --network=pentest \
    -h attacker \
    -it \
    --rm \
    --name kalibox \
    kalilinux/kali-rolling
```

В результате у вас должно получиться 2 таких терминала, в которых вы будете работать:



Кроме того, вы можете убедиться, что контейнеры подняты через команду: *docker ps*



Для проверки IP адреса, присвоенного контейнеру *kalibox* необходимо выполнить следующие команды в контейнере *kalibox*:

```
apt update
apt install net-tools
ifconfig
```

```
root@attacker: /

(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 ca:84:31:96:a0:25  txqueuelen 0  (Ethernet)
    RX packets 14237  bytes 22236913 (21.2 MiB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 9619  bytes 520680 (508.4 KiB)
    TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

С контейнером *metasploitable2* всё проще:

ip a

```
@victim: /

root@victim:/# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0@if162: <BROADCAST,MULTICAST,UP,LOWER_UP,M-DOWN> mtu 1500 qdisc noqueue
    link/ether 1a:30:7e:1e:2c:80 brd ff:ff:ff:ff:ff:ff
    inet 172.18.0.2/16 brd 172.18.255.255 scope global eth0
        valid_lft forever preferred_lft forever
root@victim:/#
```

Проверьте сетевую связность контейнеров.

Для проведения данной практической работы может понадобится тестовая учетная запись в контейнере *metasploitable2*. Добавить пользователя в контейнер *metasploitable2* можно путем выполнения следующих команд:

```
useradd user
passwd user
usermod -aG sudo user
```

Инструменты для тестирования защищенности вы можете устанавливать в контейнер *kalibox*. Чтобы установить программу *openvas* выполните следующую команду:

```
apt-get install openvas
```

Перед установкой не забудьте отключить ваш антивирус (на время установки):

```
root@attacker: /
Get:556 http://kali.download/kali kali-rolling/main amd64 gnupg-utils amd64 2.4.8-4 [194 kB]
Fetched 607 MB in 1min 16s (7958 kB/s)
E: Failed to fetch http://kali.download/kali/pool/main/i/impacket/python3-impacket_0.12.0+gite61ff5d-0kali1_all.de
b_499 Request has been forbidden by antivirus [IP: 17.253.180]
E: Unable to fetch some archives, maybe run apt update or try with --fix-missing?

(root@attacker)-[/]
```

После установки *openvas* настройте порты для работы контейнера и программы *openvas* (закройте терминал *attacker* и снова запустите):

```
sudo docker run --network=pentest \
    -h attacker \
    -it \
    --rm \
    --publish=9392:9392 \
    --name kalibox \
    kalilinux/kali-rolling
```

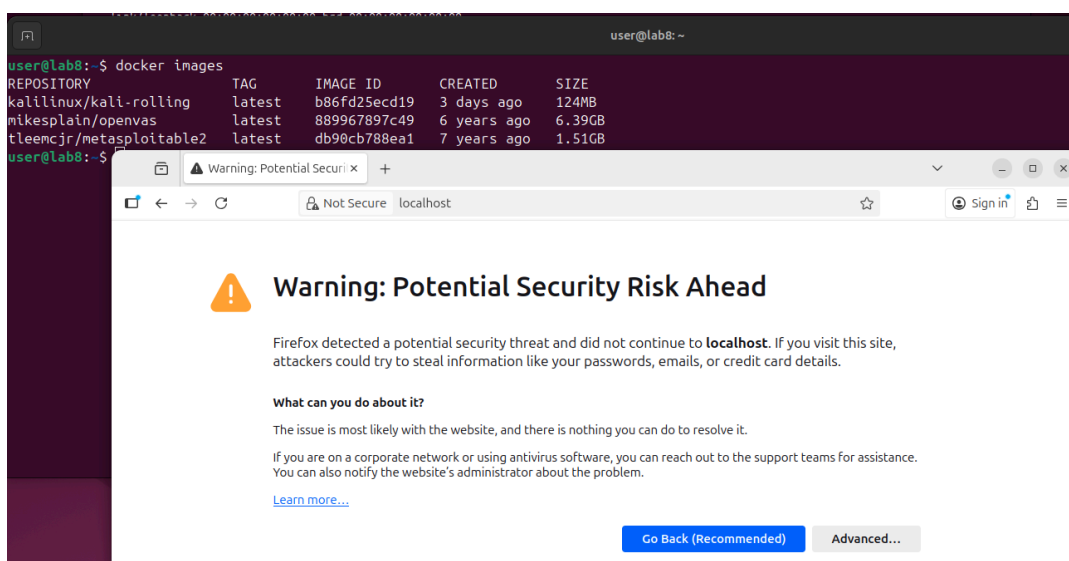
3. Работа со сканером уязвимости OpenVAS

3.1. Авторизация в OpenVAS

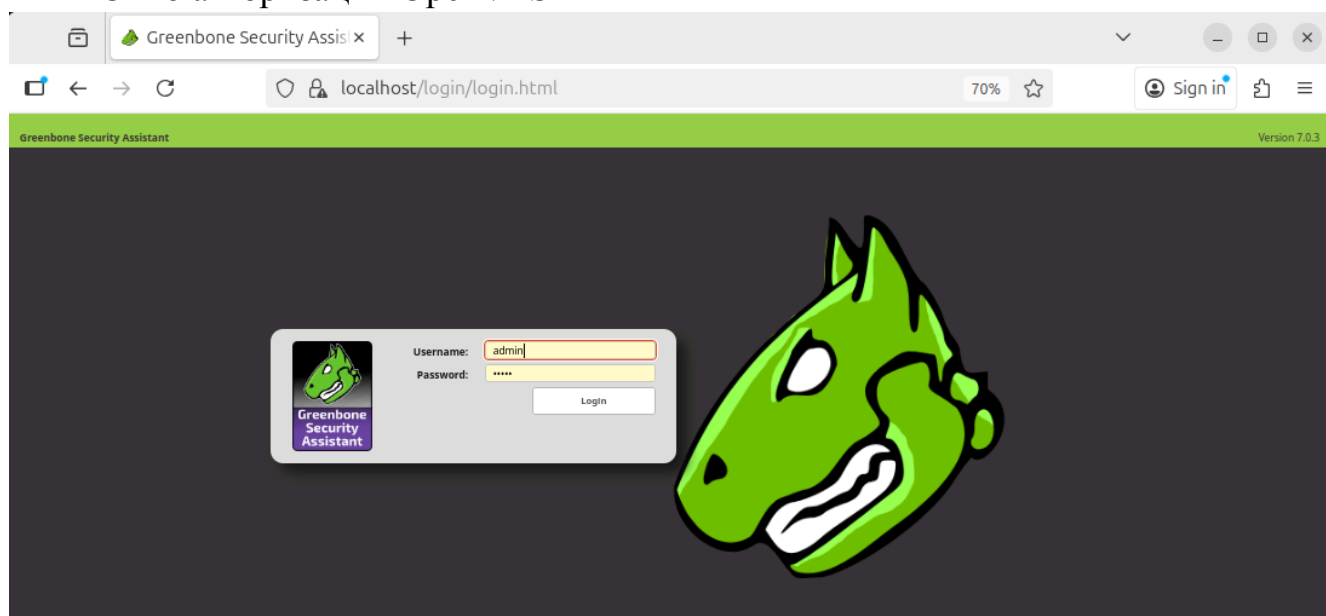
Для установки контейнера выполните команду (придется стянуть из репозитория ещё порядка 6.39GB):

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

Откройте установленный в вашей системе браузер и введите в поисковую строку адрес <https://localhost> в открывшемся окне введите логин *admin* и пароль *admin*:



Окно авторизации OpenVAS



Обновите базы используемые OpenVAS.

Краткая историческая справка.

Раньше все данные (т.н. **фиды** - наборы данных, используемые сканером для обнаружения уязвимостей) OpenVAS скачивались с сервера **feed.openvas.org**.

Этот сервер официально **закрыт и больше не поддерживается** поскольку проект **разделился**: бренд OpenVAS теперь только сканер; бренд Greenbone Community теперь представляет весь набор инструментов.

Greenbone полностью перенёс открытый фид в **новый домен** **feed.community.greenbone.net**.

Поэтому вам нужно будет изменить старый домен на новый в контейнере openvas.

Для этого выполните следующие команды:

```
docker exec -it openvas bash
```

```
## далее внутри контейнера
```

```
# Найти путь к программе, выполняющей обновления
```

```
cmd_path="$(command -v greenbone-nvt-sync)"
```

```
# Заменим старый домен на новый
```

```
sed -i -E 's/feed\.openvas\.org/feed\.community\.greenbone\.net/g' "$cmd_path"
```

```
# Проверим дополнительные конфигурационные файлы и сделаем замены
```

```
grep -RIL 'feed\.openvas\.org' /etc /usr /opt 2>/dev/null | xargs -r sed -i  
-E 's/feed\.openvas\.org/feed\.community\.greenbone\.net/g'
```

```
# Протестируем доступность фида
```

```
rsync rsync://feed\.community\.greenbone\.net/ | head
```

```
# Запускаем обновление
```

```
greenbone-nvt-sync
```

```
# перечитывает содержимое фида (NVT, SCAP, CERT) и обновляет БД  
(ООЧЕНЬ ДОЛГО)
```

```
openvasmd --rebuild --progress
```

```
# Обновляет предупреждения об инцидентах и угрозах, публикуемые CERT,  
CSIRT, # и другими структурами реагирования на киберинциденты.
```

```
greenbone-certdata-sync
```

```
# Для обновления SCAP = Security Content Automation Protocol
```

```
greenbone-scapdata-sync
```

```
# Обновляет только метаданные и ссылки в отличие от openvasmd  
--rebuild
```

```
openvasmd --update --verbose --progress
```

```
/etc/init.d/openvas-manager restart
```

```
/etc/init.d/openvas-scanner restart
```


3.2. Заведение учетной записи для проведения локальных проверок

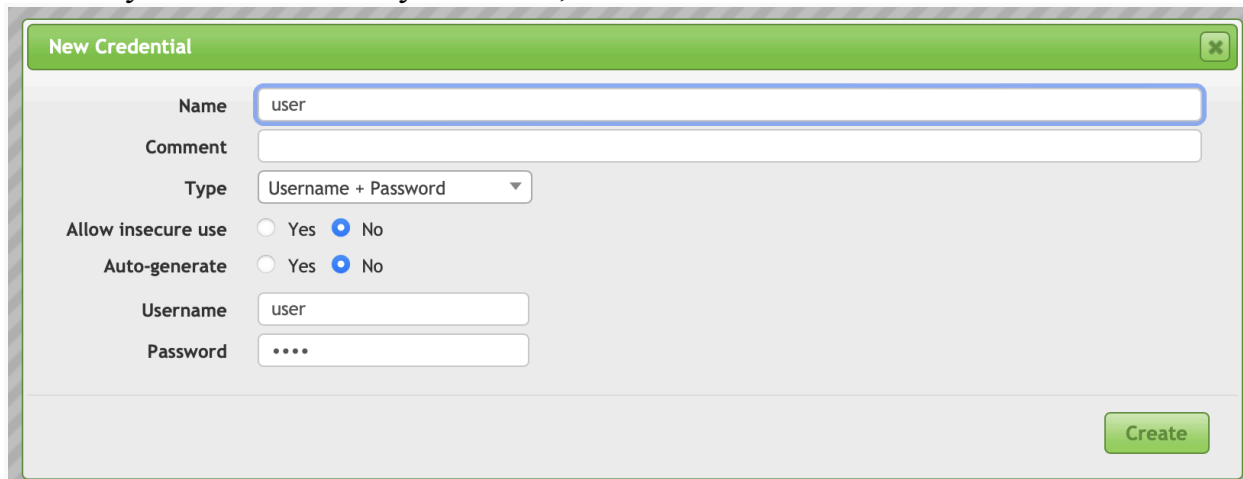
Если данный пункт сконфигурирован, то OpenVAS будет заходить на каждую машину (в нашей лабе пока это только *metasploitable2*), сканировать установленный софт, локальные настройки безопасности и выдавать алерты (предупреждения) в случае обнаружения проблем.

В этом случае время сканирования увеличивается.

Если не конфигурировать, Openvas ограничится удаленными проверками.

В разделе Configuration – Credentials

Нужно создать новую запись, нажав на значок звездочки :




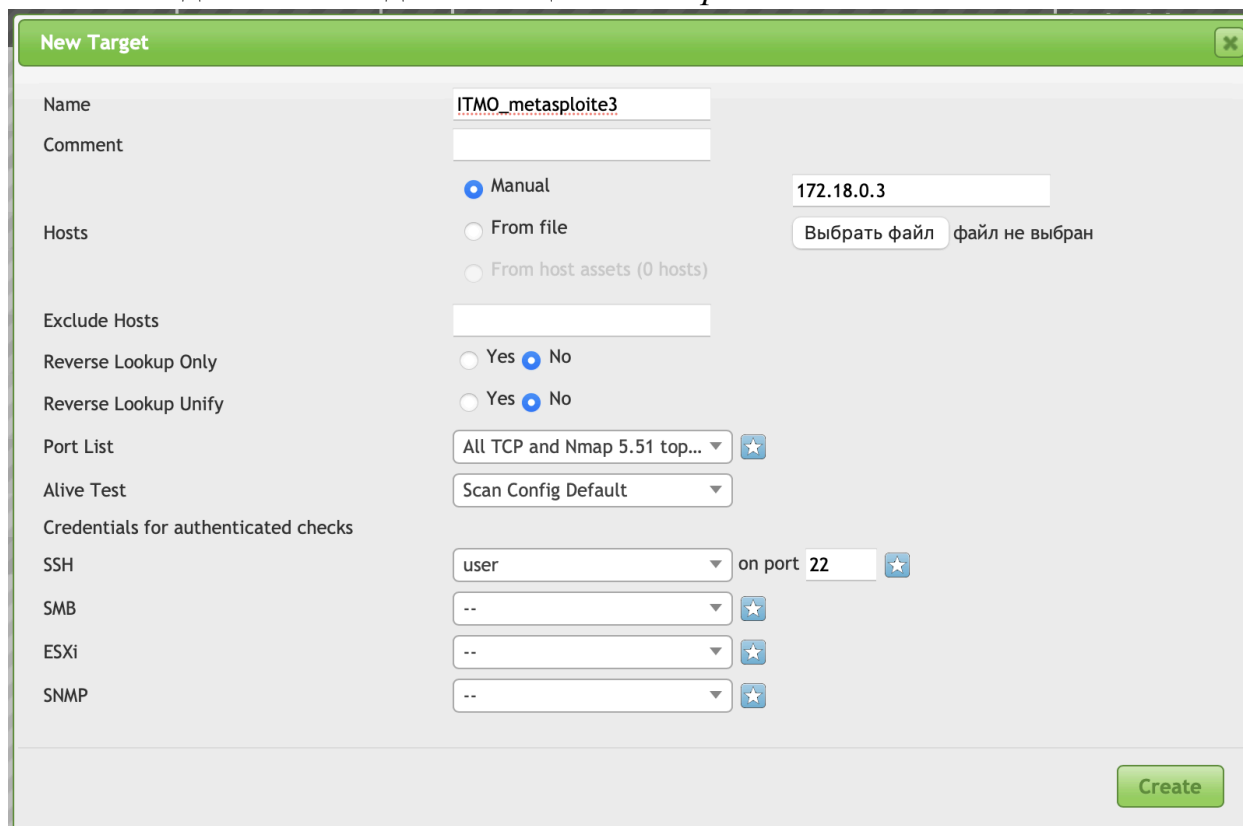
The 'New Credential' dialog box has a green header bar with a close button. It contains the following fields and controls:

- Name:** Text input with 'user' entered.
- Comment:** Empty text input.
- Type:** Dropdown menu showing 'Username + Password'.
- Allow insecure use:** Radio buttons for 'Yes' and 'No' (selected).
- Auto-generate:** Radio buttons for 'Yes' and 'No' (selected).
- Username:** Text input with 'user' entered.
- Password:** Password input field with four dots.
- Create:** Green button at the bottom right.

3.3. Выбор цели сканирования

Далее нам нужно указать диапазон адресов для сканирования и определить набор портов, которые будет проверять OpenVAS.

В разделе Configuration – Target Нужно создать новую цель, нажав на значок звездочки . Задать имя цели *metasploit2*



The 'New Target' dialog box has a green header bar with a close button. It contains the following fields and controls:

- Name:** Text input with 'ITMO_metasploit3' entered.
- Comment:** Empty text input.
- Hosts:** Radio buttons for 'Manual' (selected), 'From file', and 'From host assets (0 hosts)'. To the right of 'Manual' is a text input with '172.18.0.3' and a button 'Выбрать файл' (Choose file) with the text 'файл не выбран' (file not selected).
- Exclude Hosts:** Empty text input.
- Reverse Lookup Only:** Radio buttons for 'Yes' and 'No' (selected).
- Reverse Lookup Unify:** Radio buttons for 'Yes' and 'No' (selected).
- Port List:** Dropdown menu showing 'All TCP and Nmap 5.51 top...' with a star icon.
- Alive Test:** Dropdown menu showing 'Scan Config Default'.
- Credentials for authenticated checks:**
 - SSH:** Dropdown menu showing 'user' and 'on port 22' with a star icon.
 - SMB:** Dropdown menu showing '--' with a star icon.
 - ESXi:** Dropdown menu showing '--' with a star icon.
 - SNMP:** Dropdown menu showing '--' with a star icon.
- Create:** Green button at the bottom right.

Имя цели - ITMO_ *metasploit2*.


В разделе SSH, указан ранее созданный пользователь для проведения локальных проверок см. Примечание 2.

В разделе PortList, указывается нужный диапазон портов, в данном случае предлагаемый Nmap-ом набор популярных портов. Выбор в пользу такого диапазона, сделан в пользу оптимизации, скорости работы сканера.

В разделе Hosts указываем диапазон IP - *172.18.0.3*.

Примечание: IP-адреса в работах могут различаться. Для правильного определения IP цели используйте команду *ip a* или *ifconfig* внутри контейнера *metasploit2*

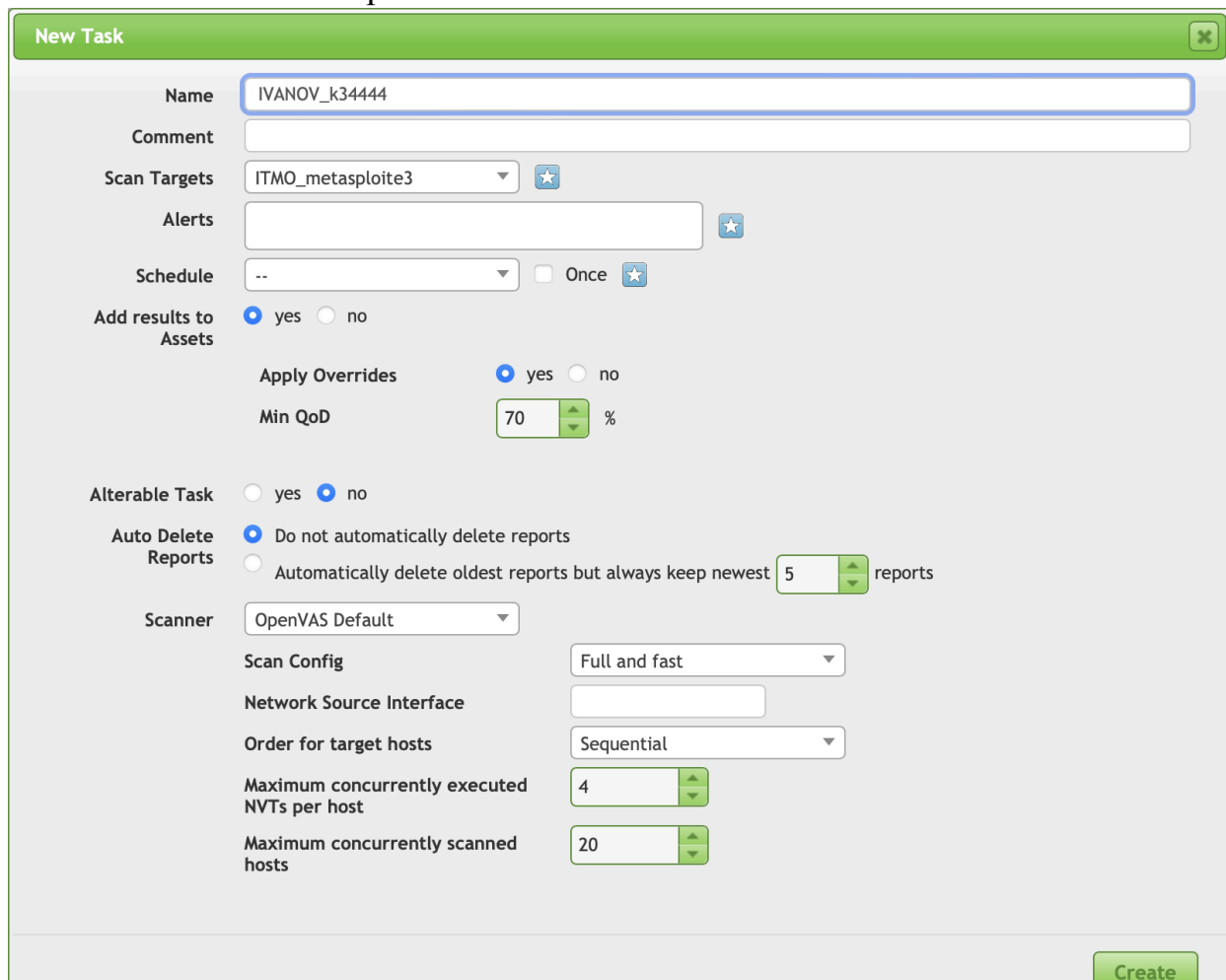
3.4. Запуск процесса сканирования

В разделе Scan — Task. Необходимо создать новую задачу, нажав на значок звездочки .

В открывшемся окне «New Task» указать имя студента и группу.

Выбрать цель сканирования в разделе Scan Targets.

Остальные настройки остаются без изменения.



Для запуска сканирования нажать на кнопку.



В конце сканирования пользователь получает отчет о выявленных уязвимостях.

Скриншот отчета о выявленных уязвимостях, необходимо указать в отчете о текущей практике.

4.3. Работа с NMAP - инструментом сканирования сетей и хостов

Запустим *kalibox*:

```
sudo docker run --network=pentest \
    -h attacker \
    -it \
    --rm \
    --name kalibox \
    kalilinux/kali-rolling
```

Установим Nmap в *kalibox*:

```
apt update
apt-get install nmap
```

Запустим *metasploitable2*:

```
sudo docker run --network=pentest \
    -h victim \
    -it \
    --rm \
    --name metasploitable2 \
    tleemcjr/metasploitable2
```

или

```
docker exec -it metasploitable2 bash
```

Выполните команду сканирования узла ИС(victim) с *kalibox*:

```
nmap -A 172.18.0.2.
```

Для удобства анализа результатов скачайте в Google Classroom файл *scan.py*, сделайте его исполняемым:

```
chmod +x scan.py
```

установите в контейнере *kalibox* пакет *python3-nmap*:

```
apt install -y python3-nmap
```

и запустите скрипт:

```
./scan.py
```

или

```
python3 scan.py
```

Результаты сканирования отразите в отчете.

!!! **Самостоятельно** выполните сканирование узла ИС со следующими параметрами и опишите результат работы:

ТСР-сканирование подключения (**-sT**) — параметр завершает трехстороннее подтверждение связи с каждым целевым портом. Если соединение установлено успешно, порт считается открытым. Трехстороннее рукопожатие — это медленный тип сканирования, поэтому, скорее всего, он будет внесен в журнал целевой машины. Данный параметр сканирования применяется по умолчанию, если Nmap запускается пользователем без привилегий.

Поскольку необходимо выполнить трехстороннее рукопожатие для каждого порта, этот тип развертки медленный и, скорее всего, будет внесен в журнал целевой машины. Если Nmap запускается пользователем без каких-либо привилегий, данный параметр сканирования выбирается по умолчанию.

SYN-сканирование (**-sS**) — также известен как полуоткрытый или скрытый SYN. С помощью этого параметра Nmap отправляет SYN-пакет, а затем ожидает ответа. Ответ SYN/ACK означает, что порт прослушивается службой, а в случае ответа RST/ACK становится ясно, что порт не прослушивается. Если ответа нет или сообщение об ошибке ICMP недоступно, порт считается фильтрованным.

Это быстрый тип сканирования. И еще одна деталь: так как трехстороннее рукопожатие не завершается, оно незаметно. Если Nmap запускается с правами привилегированного пользователя, данный параметр устанавливается по умолчанию.

TCP NULL-сканирование (**-sN**), FIN-сканирование (**-sF**), XMAS-сканирование (**-sX**) — NULL-сканирование не устанавливает все биты управления.

Сканирование FIN устанавливает только флаг FIN, а XMAS-сканирование устанавливает флаги FIN, PSH и URG. Если в ответ получен пакет RST, то порт считается закрытым, а отсутствие ответа означает, что порт открыт/отфильтрован.

TCP-сканирование Маймона (**-sM**) — такое сканирование протокола было предложено Уриэлем Маймоном (Uriel Maimon). Оно отправит пакет с установленным флагом FIN/ACK. BSD-подобные системы при открытом порте этот пакет отбросят, а если порт закрыт, будет дан ответ RST.

TCP ACK-сканирование (**-sA**) — этот тип сканирования используется для определения состояния брандмауэра и фильтрации портов. Сетевой пакет данного типа отправляет только бит ACK. Если в ответ получим RST, значит, цель не фильтруется.

TCP Window-сканирование (**-sW**) — этот тип сканирования проверяет поля TCP Window ответа первого пакета. Открытый порт выдаст положительное значение окна TCP. Закрытый порт покажет нулевое значение окна TCP.

TCP Idle-сканирование (**-sI**) — при использовании данного метода пакеты не отправляются целевой машине. Будет проведено сканирование указанного вами зомби-хоста. IDS сообщит, что атаку проводит зомби.