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 K34211

Швалов Даниил Андреевич

**Проверил**:

преподаватель практики, КТН

Назаров Михаил Сергеевич

Санкт-Петербург

2024

# Оглавление

[Введение 3](#__RefHeading___Toc6440_2741830133)

[Содержание отчета 4](#__RefHeading___Toc6442_2741830133)

[1. Установка и проверка корректности работы Docker 4](#__RefHeading___Toc6444_2741830133)

[2. Создание лаборатории для тестирования и поиска уязвимостей 5](#__RefHeading___Toc6444_2741830133_Copy_1)

[3. Работа со сканером уязвимости OpenVAS 9](#__RefHeading___Toc6444_2741830133_Copy_2)

[Вывод по работе 14](#__RefHeading___Toc7412_875968271)

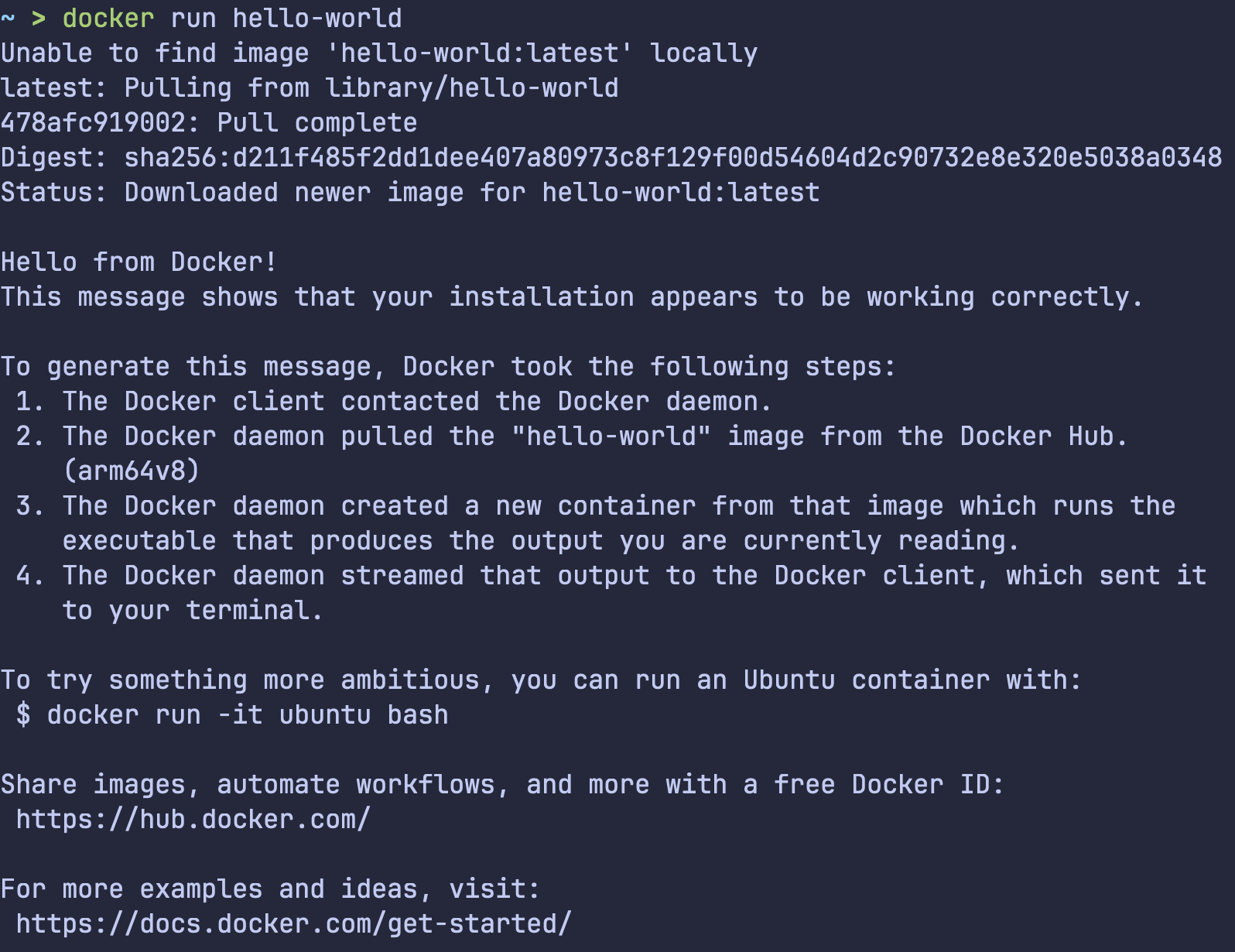
# Введение

**Цель работы**. Изучить типовой алгоритм работы с инструментами обнаружения уязвимостей информационных систем. Приобрести практические навыки по использованию сканера уязвимости. Научиться идентифицировать уязвимости информационной̆ системы.

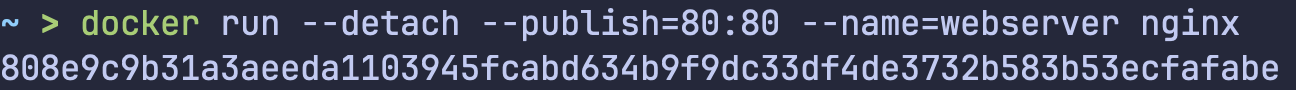
# Содержание отчета

## Установка и проверка корректности работы Docker

Для данной практической работы на компьютер был установлен Docker. Для проверки корректности его работы был загружен и запущен контейнер на основе образа «hello-world». После запуска контейнера в стандартный вывод было распечатано сообщение «Hello from Docker!», а также другая дополнительная информация. Это видно на рисунке 1.

Рисунок 1 — Запуск контейнера «hello-world»

После этого был загружен и запущен контейнер на основе образа «nginx», при этом был включен проброс 80 порта. Затем в браузере была открыта страница по адресу «localhost». На ней была отображена приветственная страница Nginx. Это продемонстрировано на рисунках 2-3.

Рисунок 2 — Запуск контейнера «nginx»

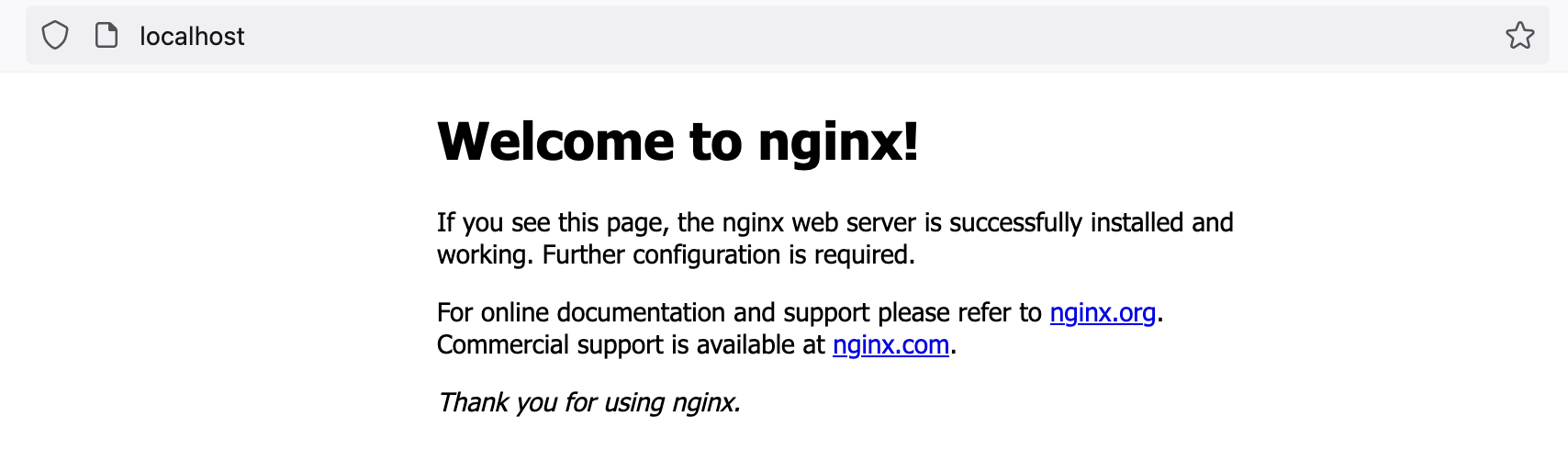
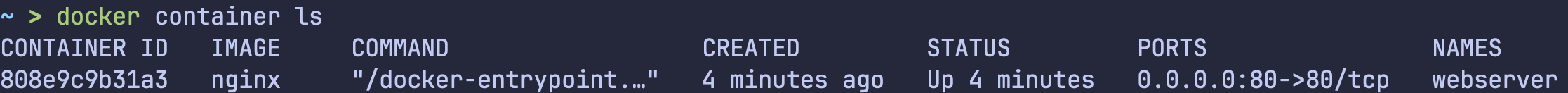


Рисунок 3 — Страница Nginx в браузере

Для отображения запущенных контейнеров была использована команда «docker container ls». В выводе команды был отображен контейнер «nginx», запущенный ранее. Это видно на рисунке 4.

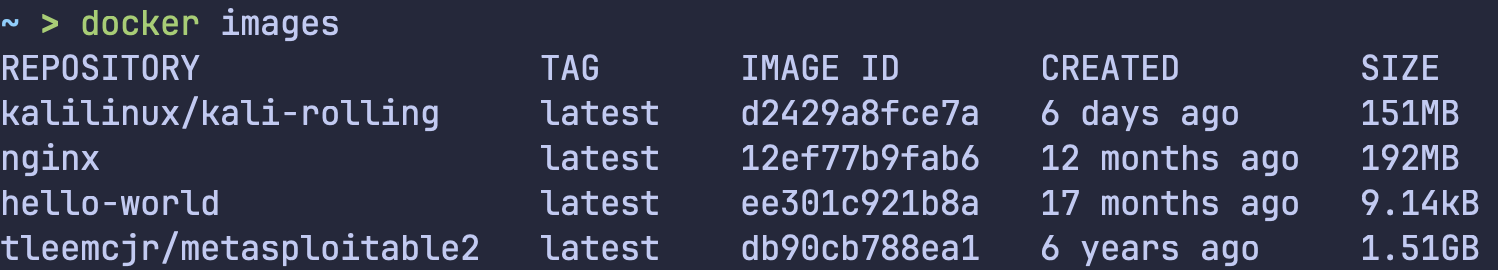
Рисунок 4 — Информация о запущенных контейнерах

## Создание лаборатории для тестирования и поиска уязвимостей

Для тестирования и поиска уязвимостей были загружены образы «tleemcjr/metasploitable2» и «kalilinux/kali-rolling». Первый образ представляет собой уязвимый сервер, а второй образ — контейнер с инструментами, используемыми для тестирования на проникновения. Эти образы были загружены с помощью команды «docker pull». Процесс загрузки ранее перечисленных образов показан на рисунках 5-6.



Рисунок 5 — Загрузка образов «metasploitable2» и «kali-rolling»

Рисунок 6 — Информация о загруженных образах

Затем была создана сеть «pentest» для контейнеров с помощью команды «docker network create». Это продемонстрировано на рисунке 7.

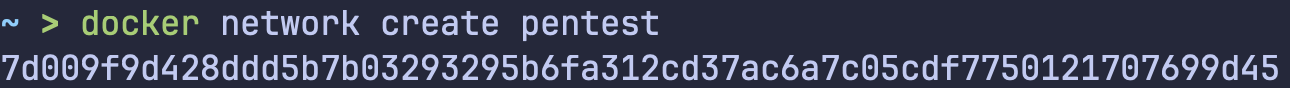


Рисунок 7 — Создание сети для контейнеров

После этого были запущены контейнеры на ранее загруженных образах. В качестве имени контейнеров были выбраны имена «metasploitable2» для образа «tleemcjr/metasploitable2» и «kalibox» для образа «kalilinux/kali-rolling». Это видно на рисунках 8-9.

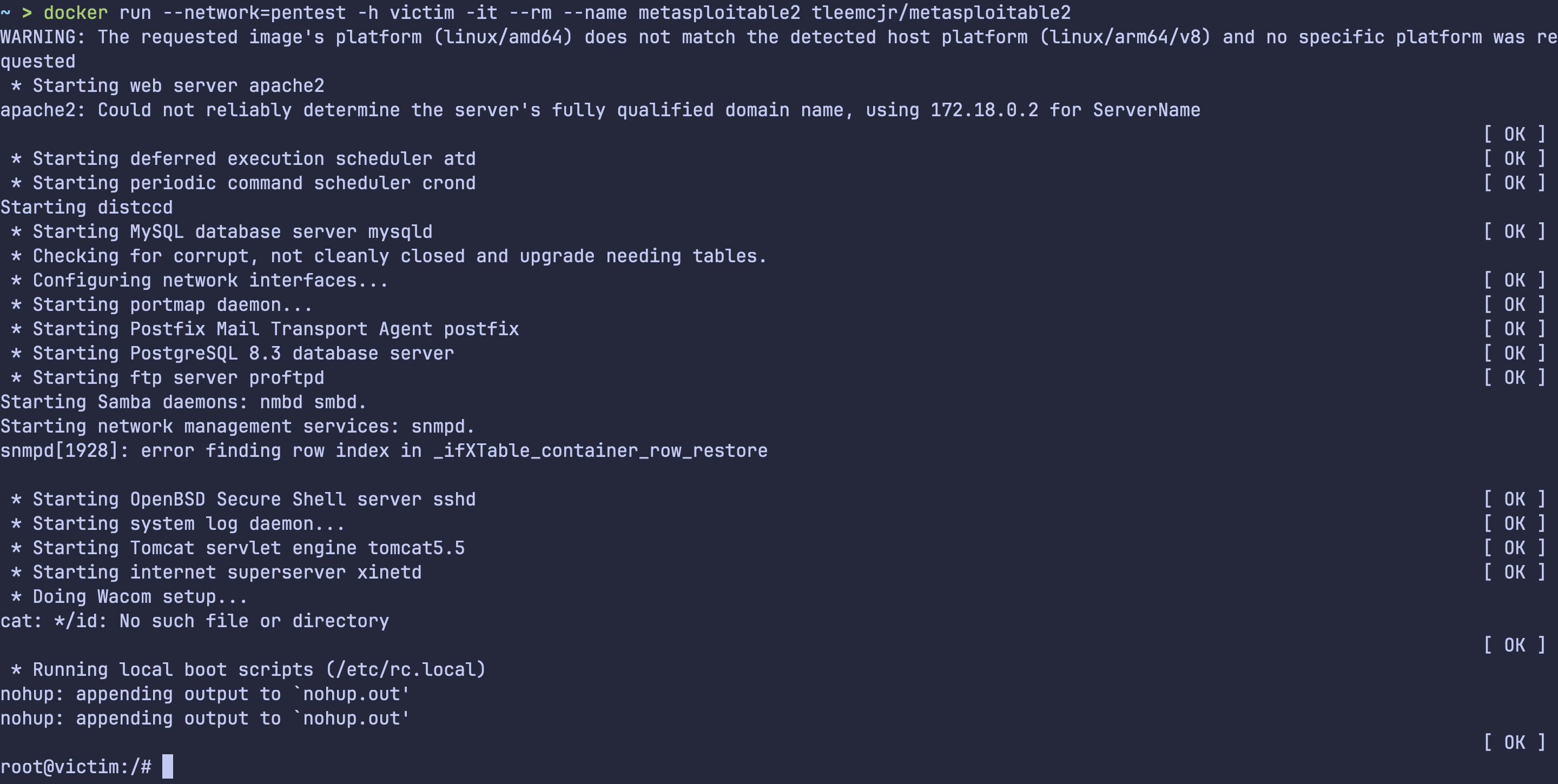


Рисунок 8 — Запуск контейнера «metasploitable2»

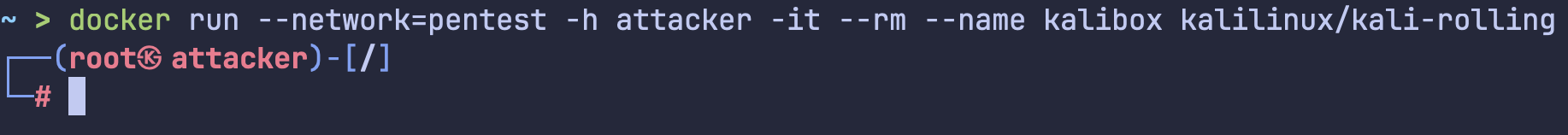


Рисунок 9 — Запуск контейнера «kalibox»

Затем была протестирована связь между контейнерами «metasploitable2» и «kalibox» по сети. Для этого была установлена утилита ifconfig. С ее помощью был получен IP-адрес контейнера «kalibox» (т. е. 172.18.0.3). После этого на контейнере «metasploitable2» с помощью утилиты ping были отправлены ICMP-запросы по адресу 172.18.0.3, т. е. по адресу контейнера «kalibox». После совершения запросов удалось получить ICMP-ответы. Таким образом, было установлено, что существует возможность передавать данные между контейнерами «kalibox» и «metasploitable2» по сети. Данный процесс показан на рисунках 10-11.

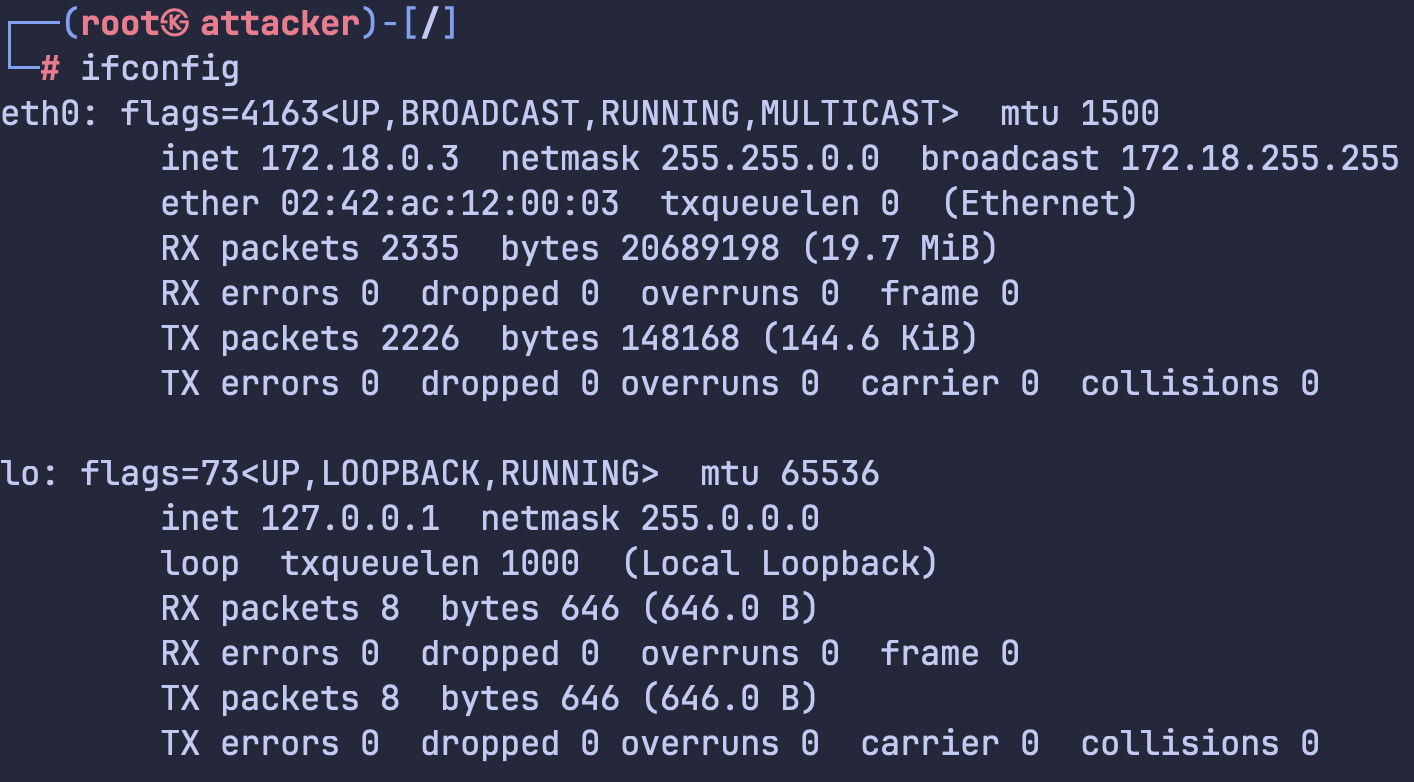


Рисунок 10 — Информация о сетевых интерфейсах на контейнере «kalibox»

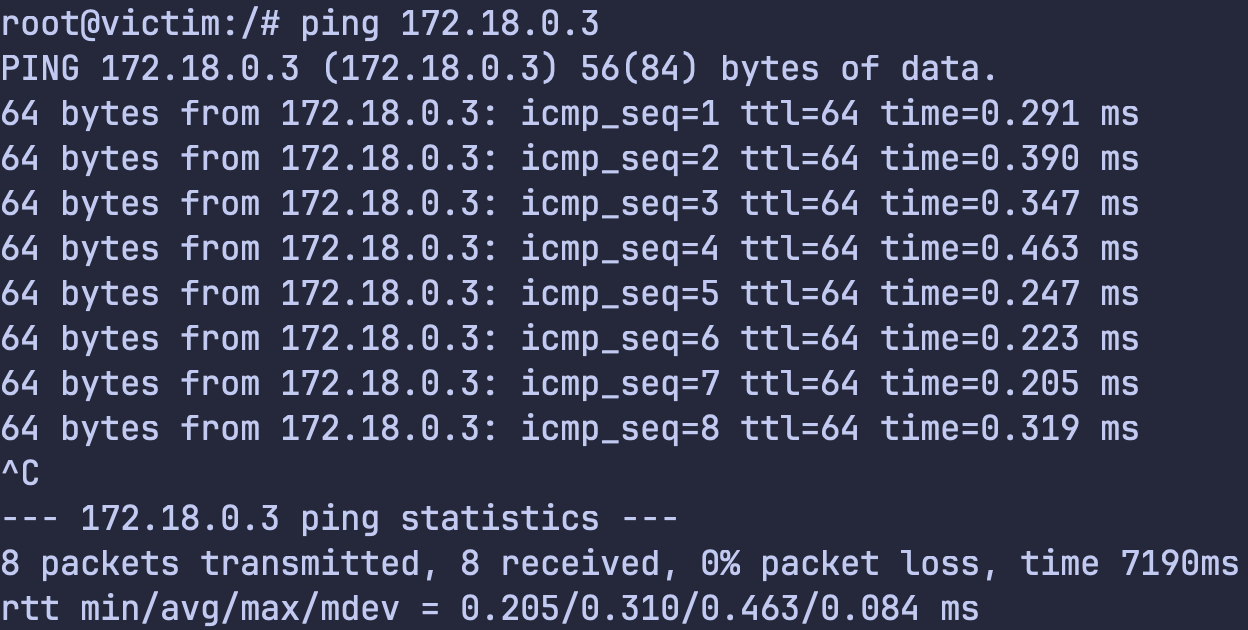


Рисунок 11 — Проверка доступности контейнера «kalibox» из контейнера «metasploitable2»

После этого на контейнере «metasploitable2» была настроена учетная запись «user». Для нее был установлен пароль, а также она была добавлена в группу «sudo». Это видно на рисунке 12.

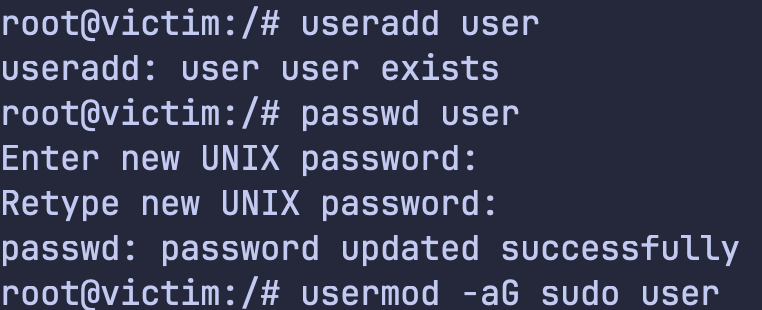


Рисунок 12 — Настройка учетной записи «user»

Затем контейнер «kalibox» был перезапущен с пробросом порта 9392. Для этого использовался флаг «--publish». Это показано на рисунке 13.

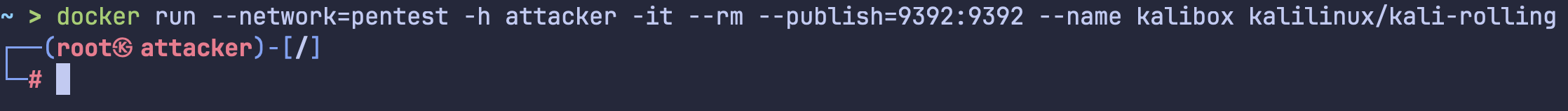


Рисунок 13 — Настройка портов контейнера «kalibox»

После этого был загружен образ «mikesplain/openvas», а на его основе был запущен контейнер «openvas» с проброшенным портом 443. Это видно на рисунке 14.

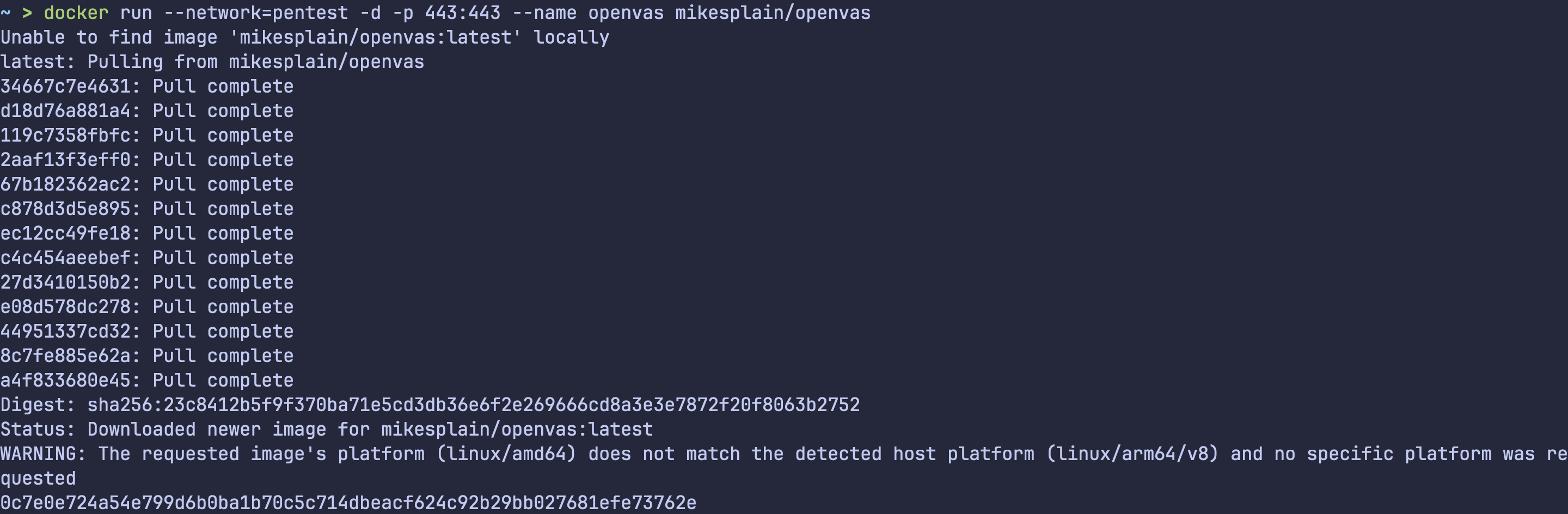


Рисунок 14 — Запуск контейнера «openvas»

## Работа со сканером уязвимости OpenVAS

Для обнаружения уязвимостей использовался сканер уязвимости OpenVAS. Как было показано выше, был запущен контейнер «openvas» с проброшенным портом 443. Благодаря этому, при открытии страницы по адресу «https://localhost» в браузере отображается страница входа в OpenVAS. После ввода логина и пароля администратора был получен доступ в OpenVAS. Страница входа показана на рисунке 15.

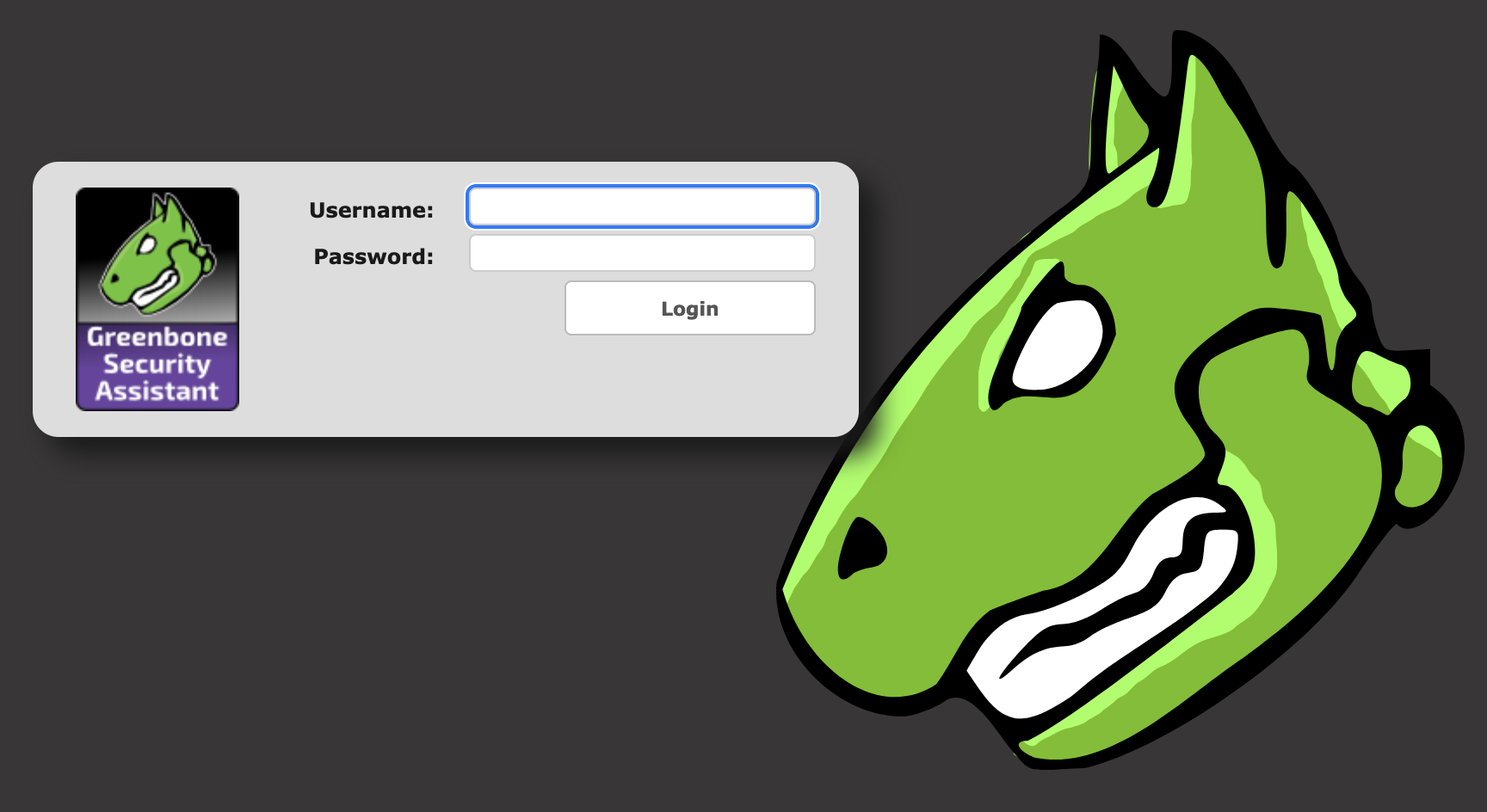
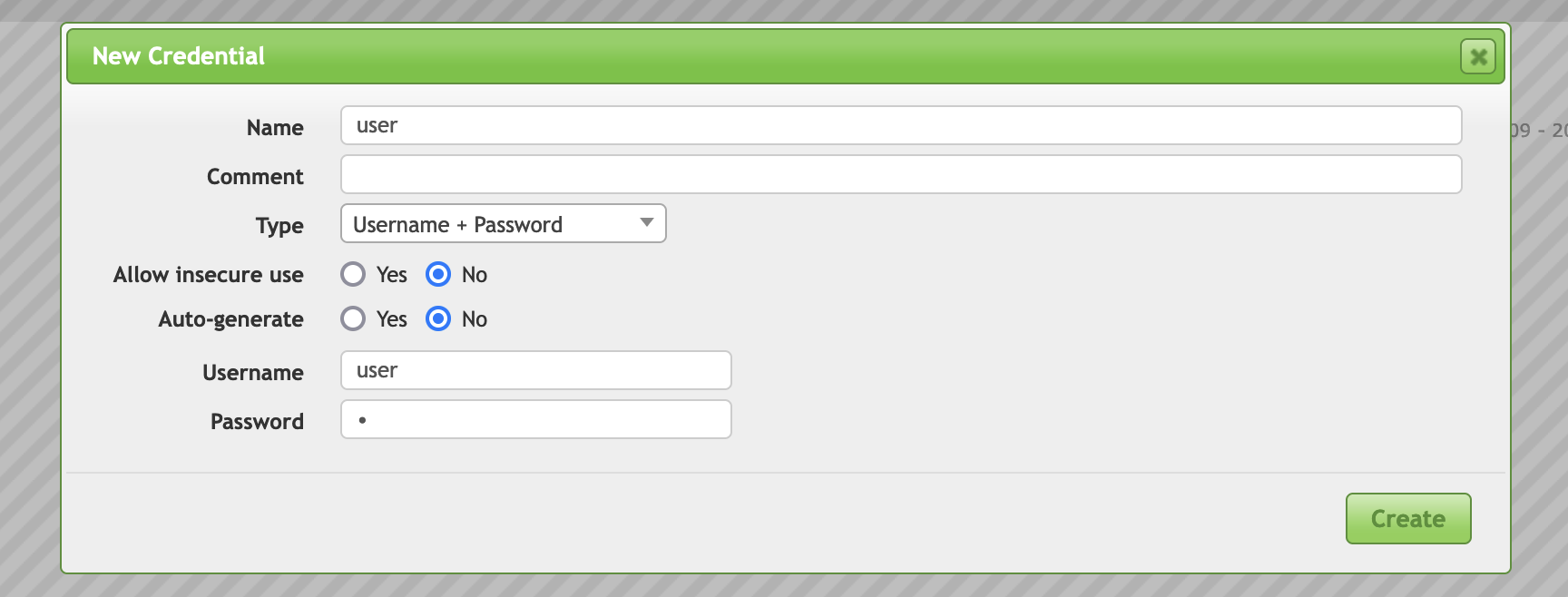


Рисунок 15 — Страница входа в веб-интерфейсе OpenVAS

После этого была заведена учетная запись для проведения локальных проверок. Для этого в разделе «Configuration — Credentials» была создана новая запись «user» с учетными данными с контейнера «metasploitable2». После создания учетной записи она появилась в списке « Credentials» Этот процесс показан на рисунках 16-17.

Рисунок 16 — Создание учетной записи в OpenVAS

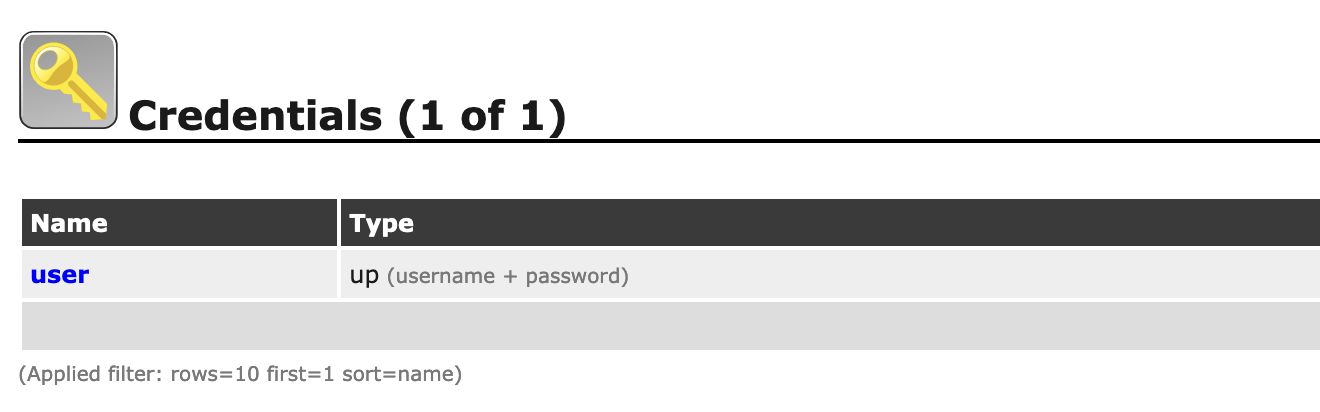


Рисунок 17 — Созданная учетная запись «user»

Затем была цель в разделе «Configuration – Target». Она необходима для указания диапазона адресов для сканирования и для определения набора портов, которые будет проверять OpenVAS. В качестве имени цели было указано имя «ITMO\_metasplot2», в качестве адреса — адрес «172.18.0.2». Также был настроен список портов в соответствии с заданием. После создания цели она появилась в списке «Targets». Это видно на рисунках 18-19

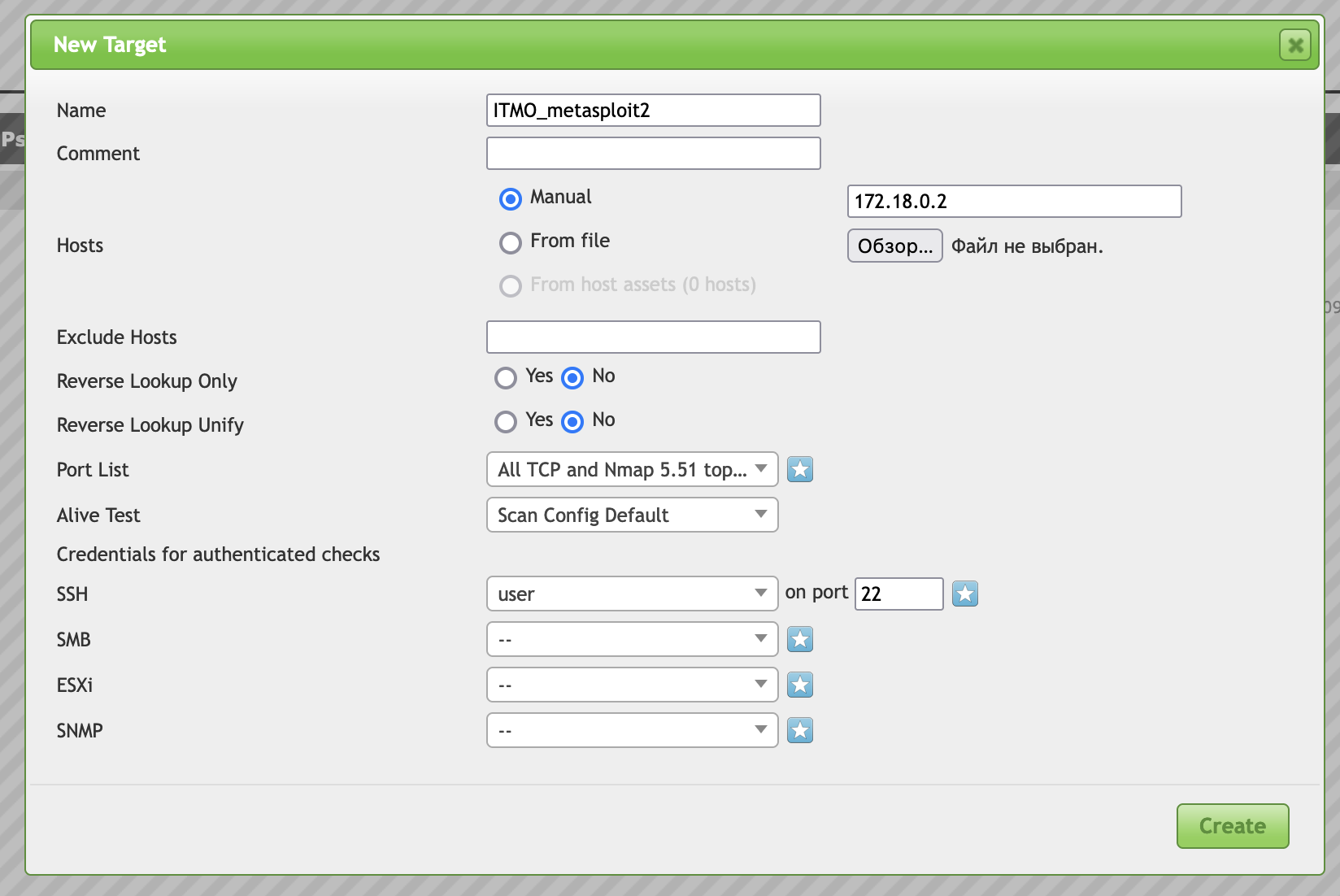


Рисунок 18 — Создание цели в OpenVAS

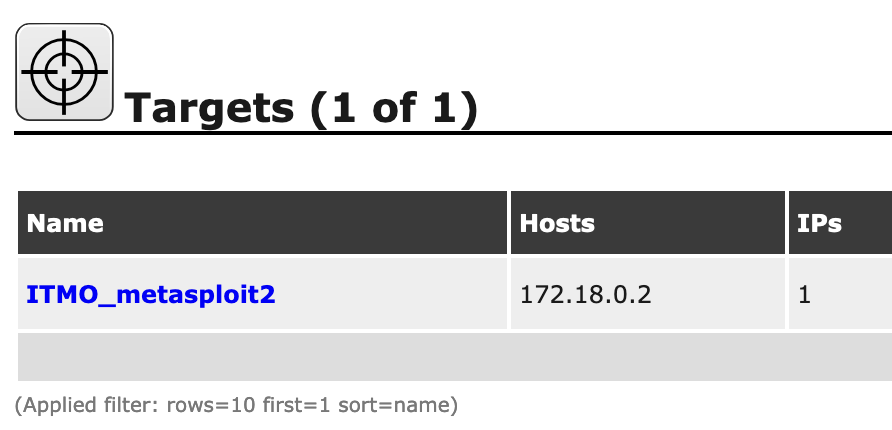


Рисунок 19 — Созданная цель «ITMO\_metasploit2»

После этого в разделе «Scan — Task» была создана задача с именем «SHVALOV\_K34211». В ней была указана ранее созданная цель «ITMO\_metasplot2». После создания задачи она появилась в списке «Tasks». Это показано на рисунках 20-21.

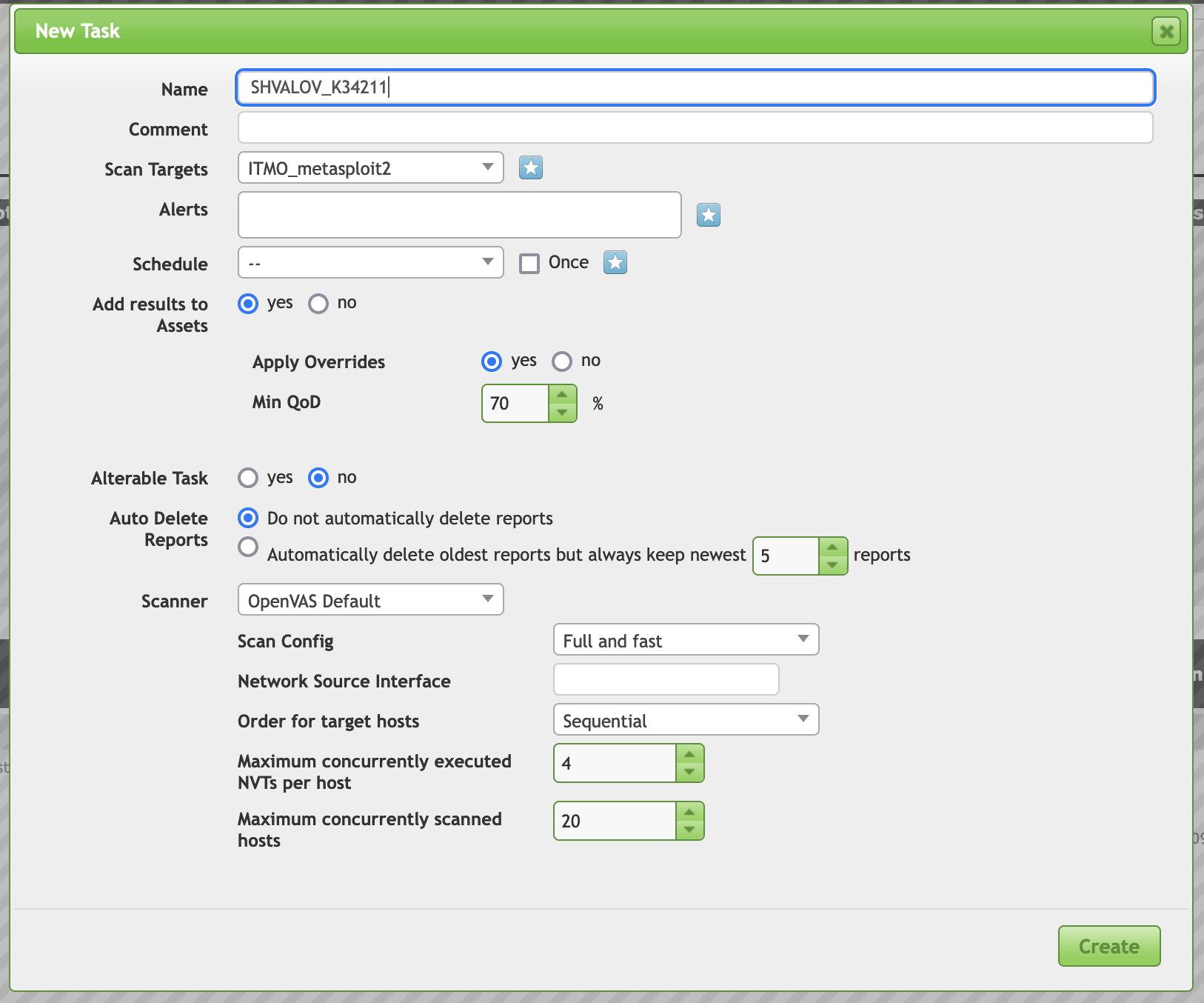
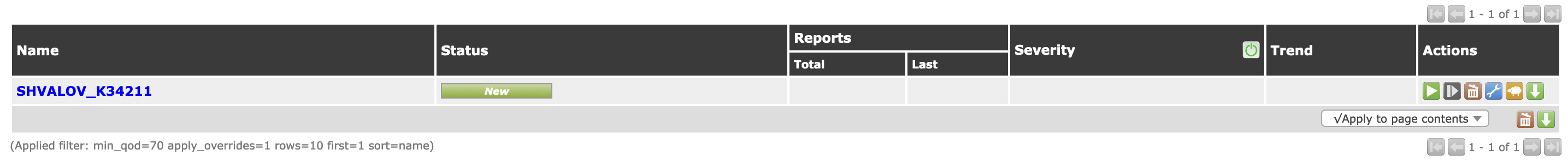
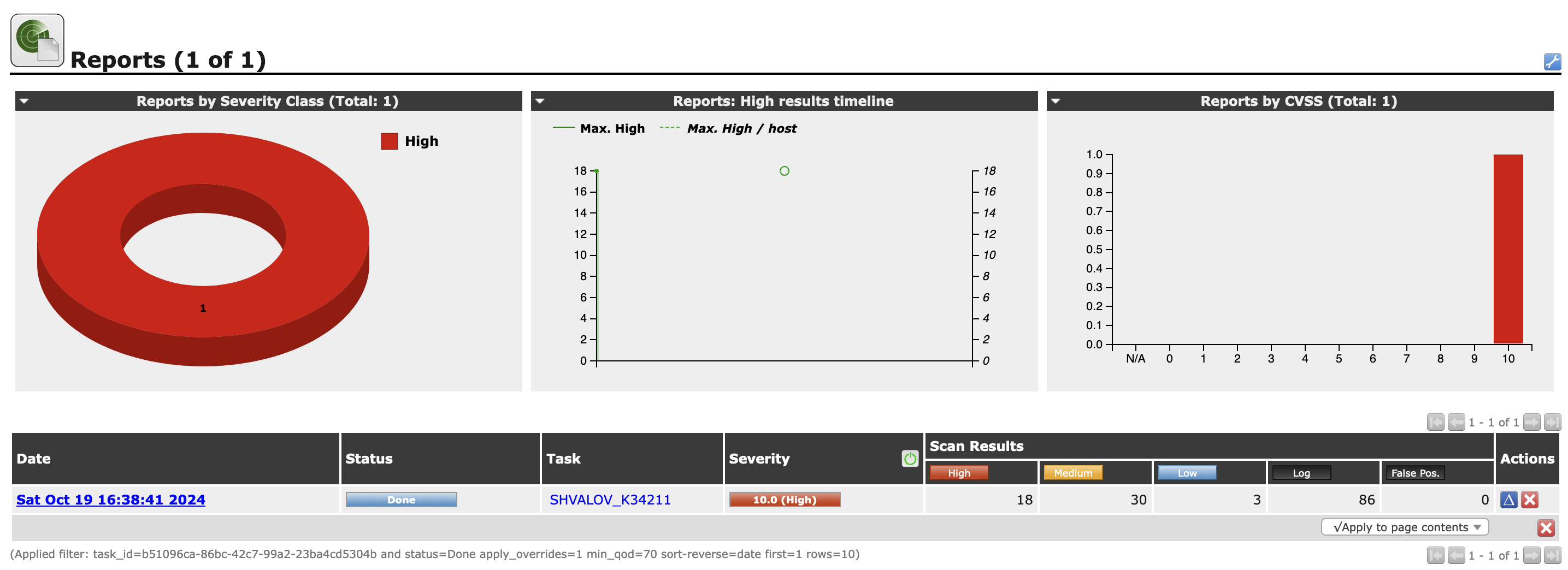


Рисунок 20 — Создание задачи в OpenVAS

Рисунок 21 — Созданная задача «SHVALOV\_K34211»

После завершения выполнения задачи был сформирован отчёт. В результатах отчёта видно, что сканируемая система содержит большое количество уязвимостей, в том числе критичных. Это продемонстрировано на рисунках 22-23.

Рисунок 22 — Сформированный отчёт после выполнения задачи

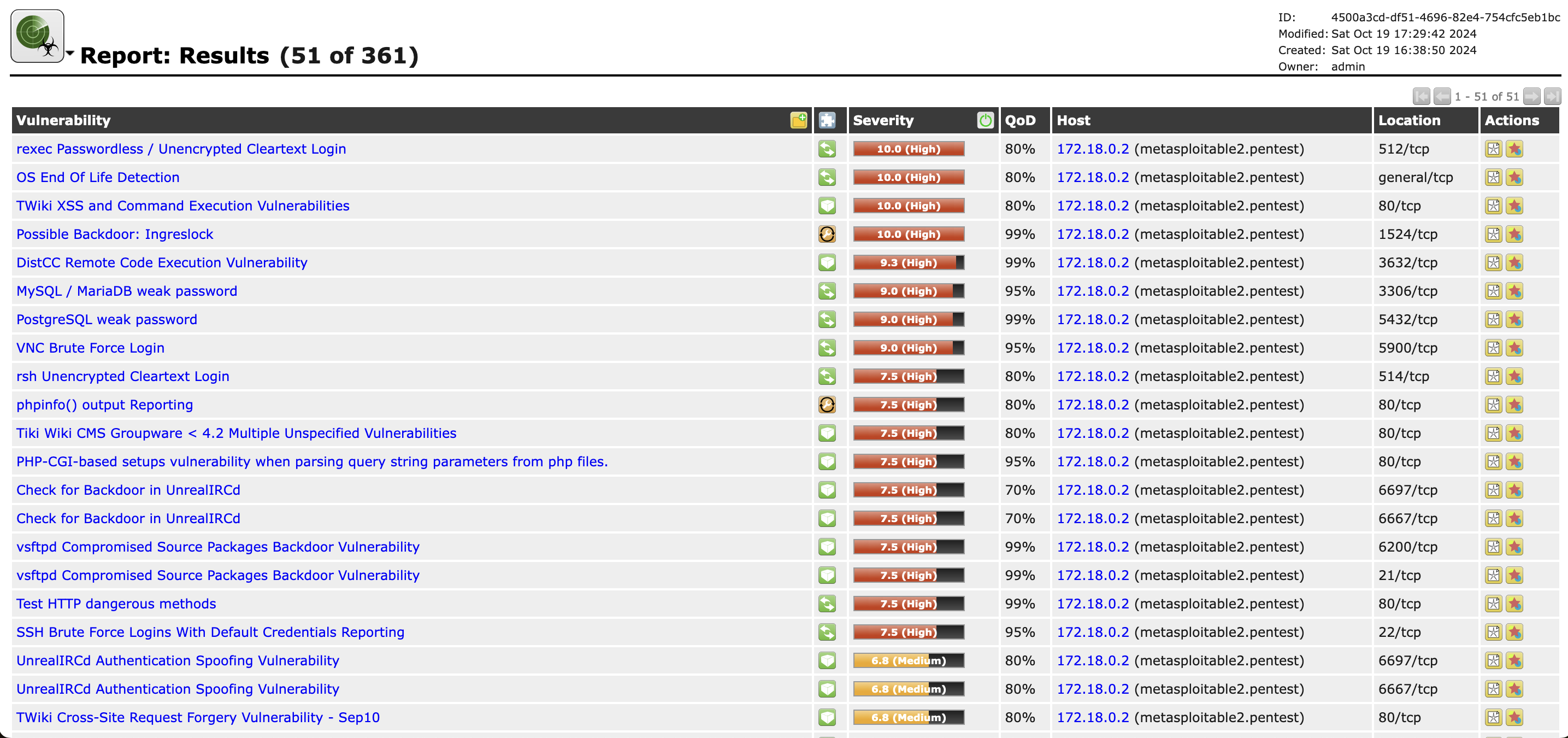


Рисунок 23 — Содержимое отчёта после выполнения задачи

# Вывод по работе

В результате выполнения данной практической работы был изучен типовой алгоритм работы с инструментами обнаружения уязвимостей информационных систем, приобретены практические навыки по использованию сканера уязвимости и идентификации уязвимости информационной̆ системы.