

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

Факультет прикладной информатики

Дисциплина:

«Основы кибербезопасности»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №8

«Обнаружение и анализ инфраструктурных уязвимостей»

Выполнил:

Швалов Даниил Андреевич, студент группы К4112с

(подпись)

Проверил:

Кравчук Алексей Владимирович, доцент практики

(отметка о выполнении)

(подпись)

Санкт-Петербург
2025 г.

СОДЕРЖАНИЕ

1 Введение.....	3
2 Ход работы.....	3
2.1 Установка Docker.....	3
2.2 Подготовка стенда.....	4
2.3 Работа со сканером уязвимости OpenVAS.....	9
3 Вывод.....	23

1 Введение

Цель работы:

- 1) изучить типовой алгоритм работы с инструментами обнаружения уязвимостей информационных систем;
- 2) приобрести практические навыки по использованию сканера инфраструктурных уязвимостей;
- 3) научиться идентифицировать уязвимости информационной системы.

2 Ход работы

2.1 Установка Docker

На устройстве, на котором выполнялась лабораторная работа, уже был установлен Docker, поэтому непосредственная установка Docker не производилась.

В начале лабораторной работы была проверена работа Docker с помощью запуска образа hello-world. Как видно на рисунке 1, контейнер с образом hello-world был успешно запущен, что говорит, что Docker работает.

```
danielshvalov > docker run --rm hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
17eec7bbc9d7: Pull complete
Digest: sha256:56433a6be3fda188089fb548eae3d91df3ed0d6589f7c2656121b911198df065
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 на основе образа hello-world

После этого также была проверена версия Docker. Как видно на рисунке 2, на устройстве установлена хоть и не последняя, но и не устаревшая версия Docker.

```
danielshvalov > docker --version
Docker version 27.2.1, build 9e34c9bb39
```

Рисунок 2 — Проверка версии Docker

2.2 Подготовка стенда

После этого с помощью команды docker pull были загружены два образа: первый, содержащий уязвимый сервер metasploitable2, и второй, содержащий Kali Linux. Это видно на рисунке 3.

```
danielshvalov > docker pull tleemcjrl/metasploitable2
Using default tag: latest
latest: Pulling from tleemcjrl/metasploitable2
7aee18c98c59: Pull complete
da9129f8f7ad: Pull complete
b1494b474174: Pull complete
84da87a98ea3: Pull complete
47fb2fc8445: Pull complete
8b6e3bfdb228: Pull complete
36d703894057: Pull complete
43cf3a9e2a40: Pull complete
Digest: sha256:e559450b37dccc1909eafa2df5b20bb052e1bd801246f4539a3ef183d5f7288a
Status: Downloaded newer image for tleemcjrl/metasploitable2:latest
docker.io/tleemcjrl/metasploitable2:latest
danielshvalov > docker pull kalilinux/kali-rolling
Using default tag: latest
latest: Pulling from kalilinux/kali-rolling
bd538341bf5d: Pull complete
Digest: sha256:c8a4573bf14f662a2d51463fd28fe6eadd47b6db458748842a754d29a06c9be3
Status: Downloaded newer image for kalilinux/kali-rolling:latest
docker.io/kalilinux/kali-rolling:latest
```

Рисунок 3 — Загрузка образов metasploitable2 и Kali Linux

После этого с помощью команды docker images были проверены образы, загруженные в Docker. Как видно на рисунке 4, ранее загруженные образы появились в Docker.

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
kalilinux/kali-rolling	latest	b86fd25ecd19	7 days ago	124MB
hello-world	latest	1b44b5a3e06a	3 months ago	10.1kB
tleemcj/r/metasploitable2	latest	db90cb788ea1	7 years ago	1.51GB

Рисунок 4 — Список загруженных образов в Docker

После этого с помощью команды docker network create была создана сеть `pentest`. Как видно на рисунке 5, сеть была успешна создана. Это видно при выполнении команды docker network ls.

```
danielshvalov > docker network create pentest
38eb97b4927d76af9b4b7f936fc81df3f5803644ef090d815d0f6392df7fbe1
danielshvalov > docker network ls
NETWORK ID      NAME      DRIVER      SCOPE
d983a4204c3b   bridge    bridge      local
e34d6964a171   host      host       local
3757ae5f6dd7   none      null       local
38eb97b4927d   pentest   bridge      local
```

Рисунок 5 — Создание сети `pentest`

После этого с помощью команды docker run в ранее созданной сети были запущены два контейнера: первый на основе образа `metasploitable2`, а второй с Kali Linux. Процесс запуска виден на рисунках 6-7.

```
danielshvalov > docker run --network=pentest -h victim -it --rm --name metasploitable2 tleemcj/r/metasploitable2
 * Starting web server apache2
apache2: Could not reliably determine the server's fully qualified domain name, using 172.18.0.2 for ServerName

 * Starting deferred execution scheduler atd
 * Starting periodic command scheduler crond
Starting distccd
 * Starting MySQL database server mysqld
 * Checking for corrupt, not cleanly closed and upgrade needing tables.
 * Configuring network interfaces...
 * Starting portmap daemon...
 * Starting Postfix Mail Transport Agent postfix
 * Starting PostgreSQL 8.3 database server
 * Starting ftp server proftpd
Starting Samba daemons: nmbd smbd.
Starting network management services: snmpd.
 * Starting OpenBSD Secure Shell server sshd
snmpd[692]: error finding row index in _ifXTable_container_row_restore

 * Starting system log daemon...
 * Starting Tomcat servlet engine tomcat5.5
 * Starting internet superserver xinetd
 * Doing Wacom setup...
 * Running local boot scripts (/etc/rc.local)
nohup: appending output to `nohup.out'
nohup: appending output to `nohup.out'

root@victim:/#
```

Рисунок 6 — Запуск контейнера на основе образа `metasploitable2`

```
danielshvalov > docker run --network=ptest -h attacker -it --rm --name kalibox kalilinux/kali-rolling
└─#
```

Рисунок 7 — Запуск контейнера с Kali Linux

После этого с помощью команды docker ps было проверено, что контейнеры действительно работают. Как видно на рисунке 8, контейнеры успешно запустились.

```
danielshvalov > docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS               NAMES
01351e422a54        kalilinux/kali-rolling   "bash"              43 seconds ago    Up 42 seconds
985a3db1b3a2        tleemcjr/metasploitable2 "sh -c '/bin/service..." About a minute ago   Up About a minute

```

Рисунок 8 — Проверка работы контейнеров с помощью команды docker ps

После этого в контейнере с Kali Linux были обновлены репозитории пакетов, а также установлен пакет net-tools. Это видно на рисунках 9-10.

```
└─# apt update
Get:1 http://mirror.krfoss.org/kali kali-rolling InRelease [34.0 kB]
Get:2 http://mirror.krfoss.org/kali kali-rolling/non-free amd64 Packages [186 kB]
Get:3 http://mirror.krfoss.org/kali kali-rolling/contrib amd64 Packages [116 kB]
Get:4 http://mirror.krfoss.org/kali kali-rolling/non-free-firmware amd64 Packages [11.3 kB]
Get:5 http://mirror.krfoss.org/kali kali-rolling/main amd64 Packages [20.9 MB]
Fetched 21.3 MB in 18s (1191 kB/s)
11 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

Рисунок 9 — Обновление репозиториев пакетов в контейнере с Kali Linux

```
└─# apt install net-tools
Installing:
 net-tools

Summary:
 Upgrading: 0, Installing: 1, Removing: 0, Not Upgrading: 11
 Download size: 194 kB
 Space needed: 939 kB / 91.7 GB available

Get:1 http://kali.download/kali kali-rolling/main amd64 net-tools amd64 2.10-2 [194 kB]
Fetched 194 kB in 1s (285 kB/s)
debconf: unable to initialize frontend: Dialog
debconf: (No usable dialog-like program is installed, so the dialog based frontend cannot be used.
debconf: falling back to frontend: Readline
debconf: unable to initialize frontend: Readline
debconf: (Can't locate Term/ReadLine.pm in @INC (you may need to install the Term::ReadLine module))
40.1 /usr/local/share/perl/5.40.1 /usr/lib/x86_64-linux-gnu/perl5/5.40 /usr/share/perl5 /usr/lib/x86_64-linux-gnu/perl/5.40 /usr/local/lib/site_perl) at /usr/share/perl5/Debconf/FrontEnd/Readline.pm line 8, <STDIN>
debconf: falling back to frontend: Teletype
Selecting previously unselected package net-tools.
(Reading database ... 5168 files and directories currently installed.)
Preparing to unpack .../net-tools_2.10-2_amd64.deb ...
Unpacking net-tools (2.10-2) ...
Setting up net-tools (2.10-2) ...
```

Рисунок 10 — Установка пакета net-tools в контейнере с Kali Linux

В пакете net-tools, установленном ранее, содержится утилита ifconfig, которая позволяет получить IP-адрес контейнера. Как видно на рисунке 11, IP-адрес контейнера с Kali Linux равен 172.18.0.3.

```
└─(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 16421 bytes 22578952 (21.5 MiB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 3181 bytes 211326 (206.3 KiB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
        loop txqueuelen 1000 (Local Loopback)
            RX packets 8 bytes 706 (706.0 B)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 8 bytes 706 (706.0 B)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Рисунок 11 — IP-адрес контейнера с Kali Linux

В контейнере с metasploitable2 ничего устанавливать не потребовалось, т. к. в данном контейнере уже есть утилита ip, которая позволяет получить IP-адрес. Как видно на рисунке 12, IP-адрес контейнера с metasploitable2 равен 172.18.0.2.

```
root@victim:/# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue qdisc 1000
    link/loopback 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
9: eth0@if10: <BROADCAST,MULTICAST,UP,LOWER_UP,M-DOWN> mtu 1500 qdisc noqueue
    link/ether 02:42:ac:12:00:02 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
```

Рисунок 12 — IP-адрес контейнера с metasploitable2

После этого для проверки сетевой связности между контейнерами в контейнере с Kali Linux был установлен пакет iputils-ping, что видно на рисунке 13.

```
└─(root㉿attacker)-[/]# apt install iputils-ping
Installing:
  iputils-ping

Installing dependencies:
  libidn2-0  libunistring5  linux-sysctl-defaults

Summary:
  Upgrading: 0, Installing: 4, Removing: 0, Not Upgrading: 11
  Download size: 644 kB
  Space needed: 2813 kB / 91.7 GB available
```

Рисунок 13 — Установка пакета iputils-ping в контейнер с Kali Linux

После установки пакета iputils-ping стала доступна утилита ping. С ее помощью, как видно на рисунке 14, была проверена доступность контейнера с metasploitable2.

```
└─(root㉿attacker)-[/]# ping 172.18.0.2
PING 172.18.0.2 (172.18.0.2) 56(84) bytes of data.
64 bytes from 172.18.0.2: icmp_seq=1 ttl=64 time=2.23 ms
64 bytes from 172.18.0.2: icmp_seq=2 ttl=64 time=1.78 ms
64 bytes from 172.18.0.2: icmp_seq=3 ttl=64 time=2.20 ms
^C
--- 172.18.0.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2007ms
rtt min/avg/max/mdev = 1.781/2.070/2.228/0.204 ms
```

Рисунок 14 — Проверка доступности контейнера с metasploitable2 из контейнера с Kali Linux

Для проверки доступности контейнера Kali Linux из контейнера metasploitable2 установка дополнительных пакетов не потребовалась. Как

видно на рисунке 15, сетевая связность между контейнерами есть в обоих контейнерах.

```
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=0.699 ms
64 bytes from 172.18.0.3: icmp_seq=2 ttl=64 time=2.90 ms
64 bytes from 172.18.0.3: icmp_seq=3 ttl=64 time=2.34 ms
^C
--- 172.18.0.3 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2005ms
rtt min/avg/max/mdev = 0.699/1.983/2.909/0.937 ms
```

Рисунок 15 — Проверка доступности контейнера с Kali Linux из контейнера с metasploitable2

После этого в контейнере с metasploitable2 была создана и настроена учетная запись user. Это видно на рисунке 16.

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

Рисунок 16 — Создание и настройка учетной записи user

2.3 Работа со сканером уязвимости OpenVAS

После этого был загружен образ и запущен контейнер, содержащий OpenVAS. Это видно на рисунке 17.

```
danielshvalov > docker run --network=pentest -d -p 443:443 --name openvas mikesplain/openvas
Unable to find image 'mikesplain/openvas:latest' locally
latest: Pulling from mikesplain/openvas
34667c7e4631: Pull complete
d18d76a881a4: Pull complete
119c7358fbfc: Pull complete
2aaf13f3eff0: Pull complete
67b182362ac2: Pull complete
c878d3d5e895: Pull complete
ec12cc49fe18: Pull complete
c4c454aeebef: Pull complete
27d3410150b2: Pull complete
e08d578dc278: Pull complete
44951337cd32: Pull complete
8c7fe885e62a: Pull complete
a4f833680e45: Pull complete
Digest: sha256:23c8412b5f9f370ba71e5cd3db36e6f2e269666cd8a3e3e7872f20f8063b2752
Status: Downloaded newer image for mikesplain/openvas:latest
69d10bc76cf45cf2715afe77c400ad9ced2a1c4ee357d7d5543677c7d648a29e
```

Рисунок 17 — Загрузка образа OpenVAS

После загрузки образа и запуска контейнера в браузере была открыта страница входа Greenbone, расположенная по адресу <https://localhost>. Внешний вид страницы представлен на рисунке 18.

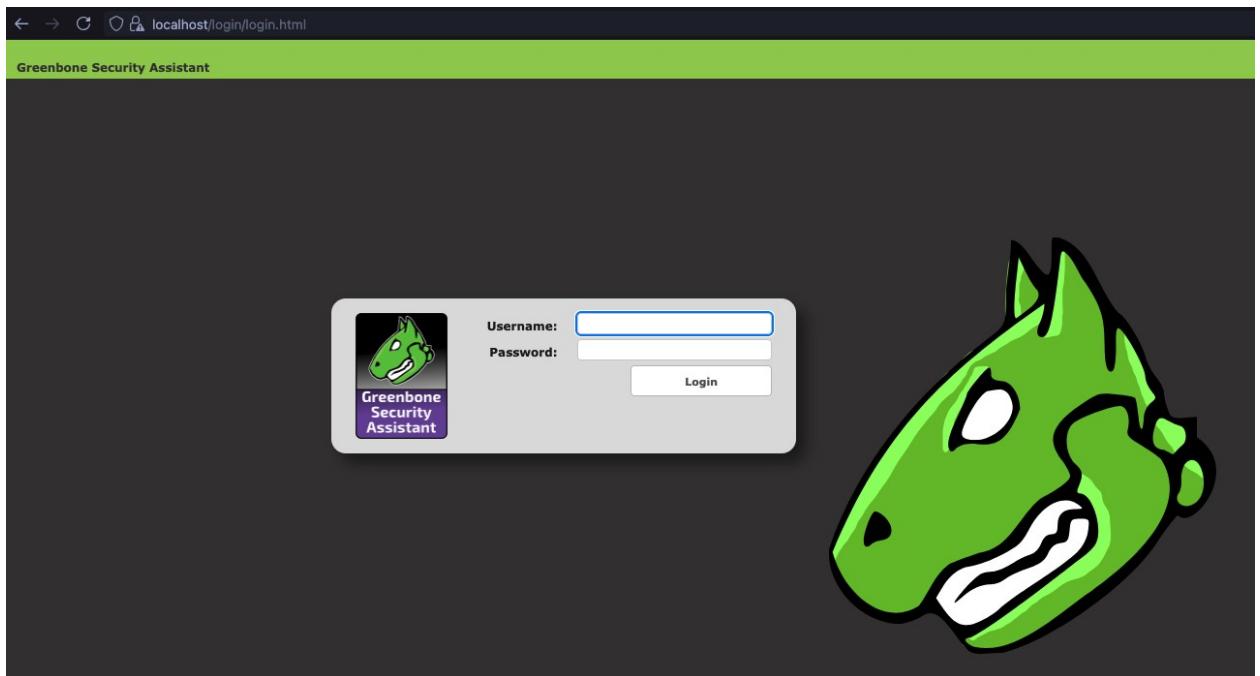


Рисунок 18 — Страница входа Greenbone

После этого в окне авторизации были введены логин admin и пароль admin. После авторизации открылась главная страница Greenbone. Это видно на рисунке 19.



Рисунок 19 — Главная страница Greenbone

После этого было выполнено подключение к контейнеру с OpenVAS. Внутри контейнера была выполнена подмена домена с OpenVAS на Greenbone. Это видно на рисунке 20.

```
root@69d10bc76cf4:/# cmd_path="$(command -v greenbone-nvt-sync)"
root@69d10bc76cf4:/# sed -i -E 's/feed\.openvas\.org/feed.community.greenbone.net/g' "$cmd_path"
root@69d10bc76cf4:/# grep -Rl 'feed\.openvas\.org' /etc /usr /opt >/dev/null | xargs -r sed -i -E 's/feed\.openvas\.org/feed.community.greenbone.net/g'
root@69d10bc76cf4:/# rsync rsync://feed.community.greenbone.net/ | head
Greenbone community feed server - http://feed.community.greenbone.net/
This service is hosted by Greenbone Networks - http://www.greenbone.net/
All transactions are logged.

If you have any questions, please use the Greenbone community portal.
See https://community.greenbone.net for details.

By using this service you agree to our terms and conditions.
```

Рисунок 20 — Замена домена OpenVAS на домен Greenbone

После этого было произведено обновление NTV, SCAP и CERT, а также обновление базы данных и метаданных. Процесс обновления представлен на рисунках 21-25.

```
root@69d10bc76cf4:/# greenbone-nvt-sync
Greenbone community feed server - http://feed.community.greenbone.net/
This service is hosted by Greenbone Networks - http://www.greenbone.net/

All transactions are logged.

If you have any questions, please use the Greenbone community portal.
See https://community.greenbone.net for details.

By using this service you agree to our terms and conditions.

Only one sync per time, otherwise the source ip will be temporarily blocked.

receiving incremental file list
plugin_feed_info.inc
      330 100% 107.42kB/s    0:00:00 (xfr#1, to-chk=0/1)

sent 43 bytes received 436 bytes 319.33 bytes/sec
total size is 330 speedup is 0.69
Greenbone community feed server - http://feed.community.greenbone.net/
This service is hosted by Greenbone Networks - http://www.greenbone.net/

All transactions are logged.

If you have any questions, please use the Greenbone community portal.
See https://community.greenbone.net for details.

By using this service you agree to our terms and conditions.

Only one sync per time, otherwise the source ip will be temporarily blocked.

receiving incremental file list
./
deleting report_formats/fae2d2f8-2a7a-11e2-b646-001f29eadec8.asc
deleting report_formats/f9d5e19c-4f90-11e4-847d-001f29e71d12.asc
deleting report_formats/f5c2a364-47d2-4700-b21d-0a7693daddab.asc
```

Рисунок 21 — Обновление NTV

```
root@69d10bc76cf4:/# openvasmd --rebuild --progress
Rebuilding NVT cache... done.
```

Рисунок 22 — Обновление базы данных

```
root@69d10bc76cf4:/# greenbone-certdata-sync
Greenbone community feed server - http://feed.community.greenbone.net/
This service is hosted by Greenbone Networks - http://www.greenbone.net/

All transactions are logged.

If you have any questions, please use the Greenbone community portal.
See https://community.greenbone.net for details.

By using this service you agree to our terms and conditions.

Only one sync per time, otherwise the source ip will be temporarily blocked.

receiving incremental file list
timestamp
    13 100%   6.35kB/s   0:00:00 (xfr#1, to-chk=0/1)
```

Рисунок 23 — Обновление CERT

```
root@69d10bc76cf4:/# greenbone-scapdata-sync
Greenbone community feed server - http://feed.community.greenbone.net/
This service is hosted by Greenbone Networks - http://www.greenbone.net/

All transactions are logged.

If you have any questions, please use the Greenbone community portal.
See https://community.greenbone.net for details.

By using this service you agree to our terms and conditions.

Only one sync per time, otherwise the source ip will be temporarily blocked.

receiving incremental file list
timestamp
    13 100%   3.17kB/s   0:00:00 (xfr#1, to-chk=0/1)

sent 43 bytes received 108 bytes  100.67 bytes/sec
total size is 13  speedup is 0.09
Greenbone community feed server - http://feed.community.greenbone.net/
This service is hosted by Greenbone Networks - http://www.greenbone.net/

All transactions are logged.

If you have any questions, please use the Greenbone community portal.
See https://community.greenbone.net for details.

By using this service you agree to our terms and conditions.

Only one sync per time, otherwise the source ip will be temporarily blocked.
```

Рисунок 24 — Обновление SCAP

```
root@69d10bc76cf4:/# openvasmd --update --verbose --progress
Updating NVT cache... done.
```

Рисунок 25 — Обновление метаданных

После этого были перезапущены менеджер и сканер OpenVAS. Это видно на рисунке 26.

```
root@69d10bc76cf4:/# /etc/init.d/openvas-manager restart
 * Restarting openvas-manager openvasmd
root@69d10bc76cf4:/# /etc/init.d/openvas-scanner restart
 * Restarting openvas-scanner openvassd
```

Рисунок 26 — Перезапуск менеджера и сканера OpenVAS

После этого в браузере с помощью меню «Configuration», «Credentials» была добавлена учетная запись user. Это видно на рисунке 27.

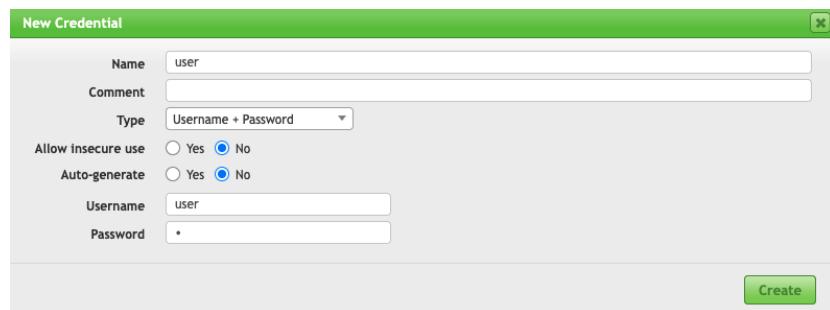


Рисунок 27 — Добавление учетной записи user в Greenbone

После этого с помощью меню «Configuration», «Target» была создана новая цель «ITMO_metasploitable2». В качестве IP-адреса цели был указан 172.18.0.2, а в качестве пользователя для SSH — учетная запись user. Это видно на рисунке 28.

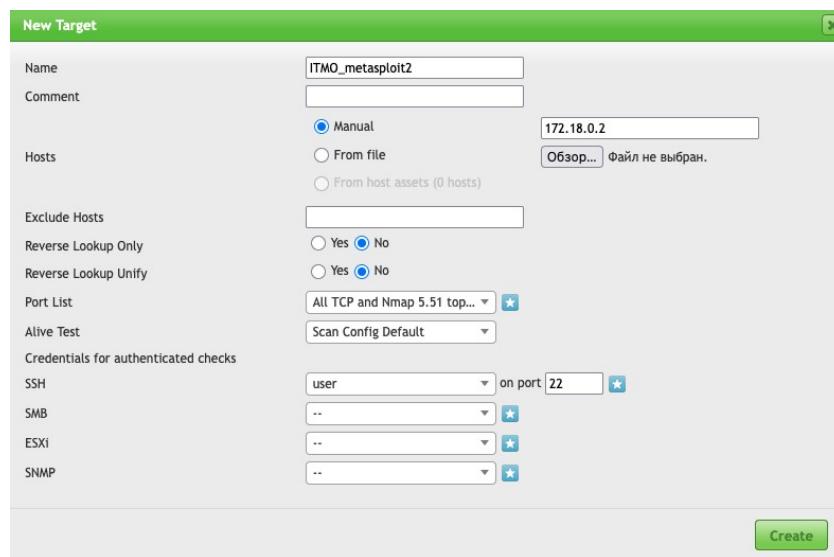


Рисунок 28 — Создание цели в Greenbone

После этого с помощью меню «Scan», «Task» была создана задача для сканирования цели. В качестве цели была выбрана ранее созданная цель, а в качестве названия была указана фамилия и номер группы. Это видно на рисунке 29.

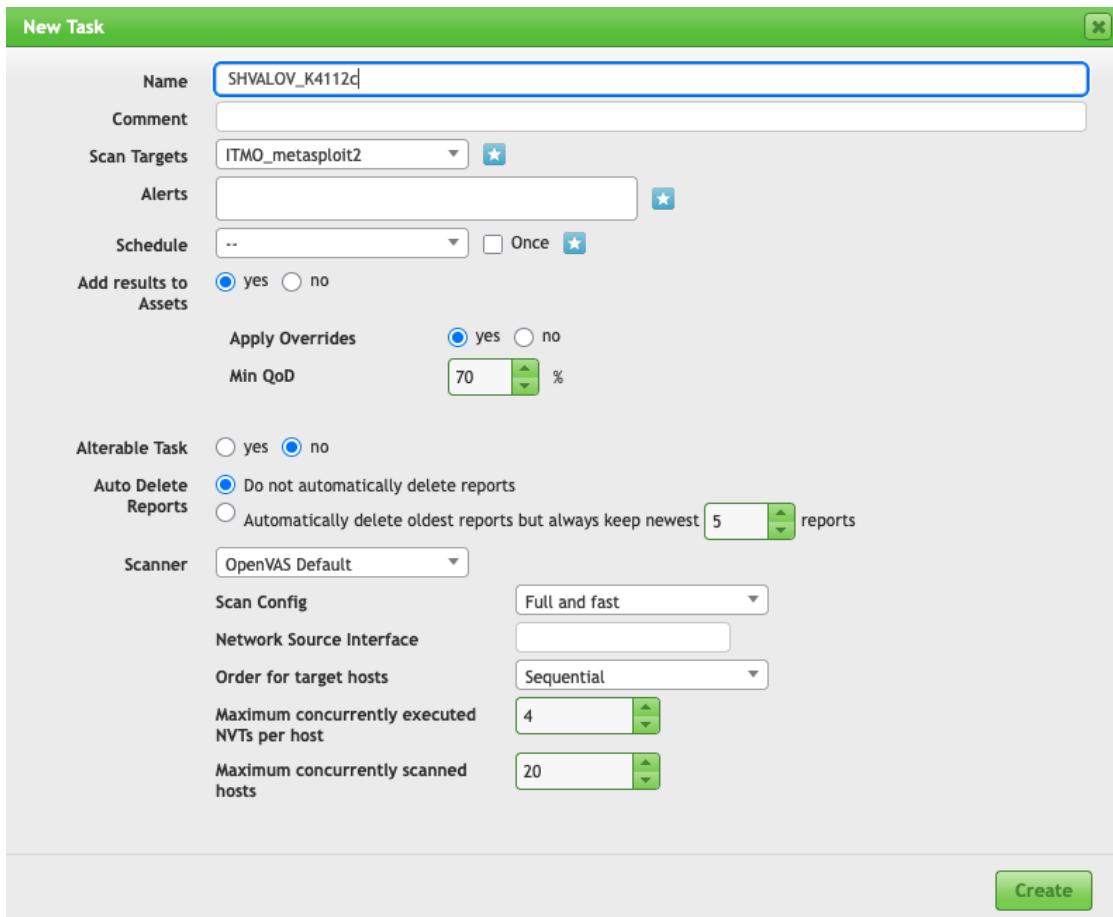


Рисунок 29 — Создание задачи для сканирования

После создания задачи было запущено сканирование metasploitable2. После завершения сканирования удалось найти 57 уязвимостей или предупреждений. Это видно на рисунке 30.

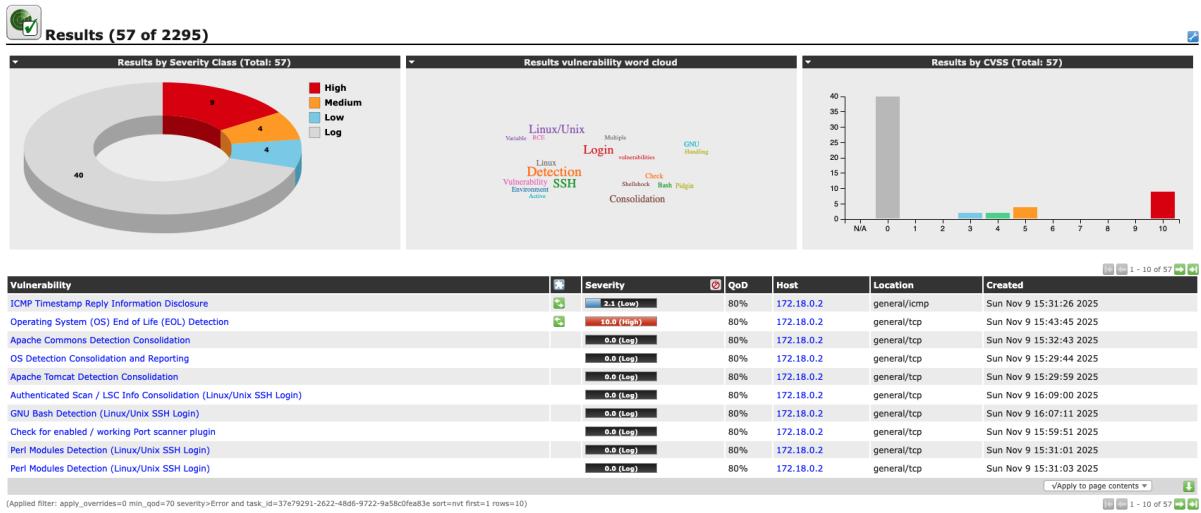


Рисунок 30 — Информация о найденных уязвимостях

Наиболее интересными уязвимостями являются:

- 1) старая версия Pidgin, которая позволяет удаленно исполнять код или приводить к падению или проблемам при работе Pidgin;
- 2) старая версия TightVNC, которая позволяет удаленно исполнять код;
- 3) старая версия Ubuntu 8.04, поддержка которой закончилась еще в 2013 году;
- 4) старая версия Bash, которая имеет различные уязвимости, позволяющие производить инъекции команд, повышение привилегий и удаленное исполнение кода;
- 5) потенциальная возможность реализации атак VMScape, TSA и SSB;
- 6) возможность получения таймстемпов через ICMP;
- 7) возможность перебора паролей для SSH;
- 8) возможность обнаружения используемого ПО: PostgreSQL, Samba, Java, Python, PHP и т. п.

После этого в контейнере с Kali Linux с помощью утилиты nmap было выполнено сканирование портов контейнера с metasploitable2. Процесс сканирования представлен на рисунке 31.

```

└─[root@attacker]─[~/]
# nmap -A 172.18.0.2
Starting Nmap 7.95 ( https://nmap.org ) at 2025-11-09 16:32 UTC
Nmap scan report for metasploitable2.pentest (172.18.0.2)
Host is up (0.0021s latency).
Not shown: 979 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          vsftpd 2.3.4
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
| ftp-syst:
|   STAT:
| FTP server status:
|     Connected to 172.18.0.3
|     Logged in as ftp
|     TYPE: ASCII
|     No session bandwidth limit
|     Session timeout in seconds is 300
|     Control connection is plain text
|     Data connections will be plain text
|     vsFTPD 2.3.4 - secure, fast, stable
|_End of status
22/tcp    open  ssh          OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
| ssh-hostkey:
|   1024 60:0f:cf:e1:c0:5f:6a:74:d6:90:24:fa:c4:d5:6c:cd (DSA)
|   2048 56:56:24:0f:21:1d:de:a7:2b:ae:61:b1:24:3d:e8:f3 (RSA)
23/tcp    open  telnet        Linux telnetd
25/tcp    open  smtp         Postfix smtpd
|_ssl-date: 2025-11-09T16:35:20+00:00; 0s from scanner time.
| ssl-cert: Subject: commonName=ubuntu804-base.localdomain/organizationName=OCOSA/stateOrProvinceName=There is no such thing outside US/countryName=XX
| Not valid before: 2010-03-17T14:07:45
| Not valid after:  2010-04-16T14:07:45
|_sslv2:
|   SSLv2 supported
| ciphers:
|   SSL2_RC4_128_EXPORT40_WITH_MD5
|   SSL2_RC4_128_WITH_MD5
|   SSL2_RC2_128_CBC_EXPORT40_WITH_MD5
|   SSL2_RC2_128_CBC_WITH_MD5
|   SSL2_DES_192_EDE3_CBC_WITH_MD5
|_SSL2_DES_64_CBC_WITH_MD5
|_smtp-commands: metasploitable.localdomain, PIPELINING, SIZE 10240000, VRFY, ETRN, STARTTLS, ENHANCEDSTATUSCODES, 8BITMIME, DSN
80/tcp    open  http         Apache httpd 2.2.8 ((Ubuntu) DAV/2)
|_http-title: Metasploitable2 - Linux
|_http-server-header: Apache/2.2.8 (Ubuntu) DAV/2

```

Рисунок 31 — Сканирование с использованием флага -A

С помощью нмар удалось обнаружить следующие порты:

- 1) 21 — порт для обмена файлами по протоколу FTP;
- 2) 22 — порт для подключения по протоколу SSH;
- 3) 23 — порт для подключения по протоколу Telnet;
- 4) 25 — порт для отправки электронных писем по протоколу SMTP;
- 5) 80 — порт для передачи HTTP;
- 6) 111 — порт для преобразования RPC-запросов;
- 7) 139 — порт для работы с файлами по протоколу Samba;
- 8) 445 — порт для работы с файлами по протоколу Samba;
- 9) 512 — порт для удаленного выполнения команд;
- 10) 513 — порт для удаленного входа в систему;
- 11) 514 — порт, на котором работает tcpwrapper;
- 12) 1099 — порт для работы с реестром RMI;
- 13) 1524 — порт для управления СУБД Ingres;

- 14) 2121 — порт для обмена файлами по протоколу FTP;
- 15) 3306 — порт для подключения к СУБД MySQL;
- 16) 5432 — порт для подключения к СУБД PostgreSQL;
- 17) 5900 — порт для удаленного доступа к рабочему столу;
- 18) 6000 — порт для подключения к X11 серверу;
- 19) 6667 — порт для работы IRC;
- 20) 8009 — порт для работы Apache Jserv;
- 21) 8180 — порт для работы Apache Tomcat.

Также с помощью nmap удалось получить различную дополнительную информацию о хосте, такую как ОС, версию ядра Linux, MAC-адрес и т. п. Это видно на рисунке 32.

```

MAC Address: 02:42:AC:12:00:02 (Unknown)
Device type: general purpose|router
Running: Linux 4.X|5.X, MikroTik RouterOS 7.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5 cpe:/o:mikrotik:routeros:7 cpe:/o:linux:linux_kernel:5.6.3
OS details: Linux 4.15 - 5.19, OpenWrt 21.02 (Linux 5.4), MikroTik RouterOS 7.2 - 7.5 (Linux 5.6.3)
Network Distance: 1 hop
Service Info: Hosts: metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
|_clock-skew: mean: 1h15m00s, deviation: 2h30m01s, median: 0s
| smb-os-discovery:
|   OS: Unix (Samba 3.0.20-Debian)
|   Computer name: victim
|   NetBIOS computer name:
|   Domain name:
|   FQDN: victim
|_ System time: 2025-11-09T11:35:13-05:00
|_nbstat: NetBIOS name: VICTIM, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_ message_signing: disabled (dangerous, but default)
|_smb2-time: Protocol negotiation failed (SMB2)

TRACEROUTE
HOP RTT      ADDRESS
1  2.12 ms metasploitable2.pentest (172.18.0.2)

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/. .
Nmap done: 1 IP address (1 host up) scanned in 166.32 seconds

```

Рисунок 32 — Информация об ОС, FQDN, MAC-адресе

После этого в контейнер с Kali Linux был загружен Python-скрипт scan.py, позволяющий представить вывод nmap в более удобном виде. После этого для скрипта было выдано разрешение на исполнение, что видно на рисунке 33.

```
(root@attacker)-[~]
# chmod +x scan.py
```

Рисунок 33 — Выдача разрешения на исполнение scan.py

Для работы скрипта scan.py требуется Python-библиотека nmap, поэтому она была установлена с помощью apt. Это видно на рисунке 34.

```
(root@attacker)-[~]
# apt install -y python3-nmap
Installing:
  python3-nmap

Summary:
  Upgrading: 0, Installing: 1, Removing: 0, Not Upgrading: 11
  Download size: 25.5 kB
  Space needed: 113 kB / 73.5 GB available
```

Рисунок 34 — Установка python3-nmap

После этого скрипт scan.py был запущен с флагом -A. Как видно на рисунке 35, результаты остались теми же, что и прежде.

```
(root@attacker)-[~]
# ./scan.py
Scanning 172.18.0.2 args '-A' ...

Host: 172.18.0.2 (metasploitable2.pentest) State: up
  21/tcp open  ftp vsftpd 2.3.4
  22/tcp open  ssh OpenSSH 4.7p1 Debian 8ubuntu1
  23/tcp open  telnet Linux telnetd
  25/tcp open  smtp Postfix smtpd
  80/tcp open  http Apache httpd 2.2.8
  111/tcp open  rpcbind 2
  139/tcp open  netbios-ssn Samba smbd 3.X - 4.X
  445/tcp open  netbios-ssn Samba smbd 3.0.20-Debian
  512/tcp open  exec
  513/tcp open  login
  514/tcp open  tcpwrapped
  1099/tcp open  java-rmi GNU Classpath grmiregistry
  1524/tcp open  ingreslock
  2121/tcp open  ftp ProFTPD 1.3.1
  3306/tcp open  mysql MySQL 5.0.51a-3ubuntu5
  5432/tcp open  postgresql PostgreSQL DB 8.3.0 - 8.3.7
  5900/tcp open  vnc VNC
  6000/tcp open  X11
  6667/tcp open  irc UnrealIRCd
  8009/tcp open  ajp13 Apache Jserv
  8180/tcp open  http Apache Tomcat/Coyote JSP engine 1.1
OS guesses:
- Linux 4.15 - 5.19 (accuracy 100%)
- OpenWrt 21.02 (Linux 5.4) (accuracy 100%)
- MikroTik RouterOS 7.2 - 7.5 (Linux 5.6.3) (accuracy 100%)
```

Рисунок 35 — Сканирование с использованием флага -A

После этого было выполнено сканирование с использованием флагом -sT и -sS. Как видно на рисунках 36-37, результаты остались те же.

```
└─(root㉿attacker)-[~/]
# ./scan.py
Scanning 172.18.0.2 args '-sT' ...

Host: 172.18.0.2 (metasploitable2.pentest)  State: up
 21/tcp open  ftp
 22/tcp open  ssh
 23/tcp open  telnet
 25/tcp open  smtp
 80/tcp open  http
 111/tcp open  rpcbind
 139/tcp open  netbios-ssn
 445/tcp open  microsoft-ds
 512/tcp open  exec
 513/tcp open  login
 514/tcp open  shell
 1099/tcp open  rmiregistry
 1524/tcp open  ingreslock
 2121/tcp open  ccproxy-ftp
 3306/tcp open  mysql
 5432/tcp open  postgresql
 5900/tcp open  vnc
 6000/tcp open  X11
 6667/tcp open  irc
 8009/tcp open  ajp13
 8180/tcp open
```

Рисунок 36 — Сканирование с использованием флага -sT

```
└─(root㉿attacker)-[~/]
# ./scan.py
Scanning 172.18.0.2 args '-sS' ...

Host: 172.18.0.2 (metasploitable2.pentest)  State: up
 21/tcp open  ftp
 22/tcp open  ssh
 23/tcp open  telnet
 25/tcp open  smtp
 80/tcp open  http
 111/tcp open  rpcbind
 139/tcp open  netbios-ssn
 445/tcp open  microsoft-ds
 512/tcp open  exec
 513/tcp open  login
 514/tcp open  shell
 1099/tcp open  rmiregistry
 1524/tcp open  ingreslock
 2121/tcp open  ccproxy-ftp
 3306/tcp open  mysql
 5432/tcp open  postgresql
 5900/tcp open  vnc
 6000/tcp open  X11
 6667/tcp open  irc
 8009/tcp open  ajp13
 8180/tcp open
```

Рисунок 37 — Сканирование с использованием флага -sS

После этого было выполнено сканирование с использованием флагов -sN, -sF и -sX. Как видно на рисунках 38-40, в таком случае новых портов найти также не удалось, т. е. были найдены все те же порты, что и прежде.

```
[root@attacker]# ./scan.py
Scanning 172.18.0.2 args '-sN' ...

Host: 172.18.0.2 (metasploitable2.pentest)  State: up
 21/tcp open|filtered ftp
 22/tcp open|filtered ssh
 23/tcp open|filtered telnet
 25/tcp open|filtered smtp
 80/tcp open|filtered http
 111/tcp open|filtered rpcbind
 139/tcp open|filtered netbios-ssn
 445/tcp open|filtered microsoft-ds
 512/tcp open|filtered exec
 513/tcp open|filtered login
 514/tcp open|filtered shell
 1099/tcp open|filtered rmiregistry
 1524/tcp open|filtered ingreslock
 2121/tcp open|filtered ccproxy-ftp
 3306/tcp open|filtered mysql
 5432/tcp open|filtered postgresql
 5900/tcp open|filtered vnc
 6000/tcp open|filtered X11
 6667/tcp open|filtered irc
 8009/tcp open|filtered ajp13
 8180/tcp open|filtered
```

Рисунок 38 — Сканирование с использованием флага -sN

```
[root@attacker]# ./scan.py
Scanning 172.18.0.2 args '-sF' ...

Host: 172.18.0.2 (metasploitable2.pentest)  State: up
 21/tcp open|filtered ftp
 22/tcp open|filtered ssh
 23/tcp open|filtered telnet
 25/tcp open|filtered smtp
 80/tcp open|filtered http
 111/tcp open|filtered rpcbind
 139/tcp open|filtered netbios-ssn
 445/tcp open|filtered microsoft-ds
 512/tcp open|filtered exec
 513/tcp open|filtered login
 514/tcp open|filtered shell
 1099/tcp open|filtered rmiregistry
 1524/tcp open|filtered ingreslock
 2121/tcp open|filtered ccproxy-ftp
 3306/tcp open|filtered mysql
 5432/tcp open|filtered postgresql
 5900/tcp open|filtered vnc
 6000/tcp open|filtered X11
 6667/tcp open|filtered irc
 8009/tcp open|filtered ajp13
 8180/tcp open|filtered
```

Рисунок 39 — Сканирование с использованием флага -sF

```
(root@attacker)-[~]
# ./scan.py
Scanning 172.18.0.2 args '-sX' ...

Host: 172.18.0.2 (metasploitable2.pentest) State: up
21/tcp open|filtered ftp
22/tcp open|filtered ssh
23/tcp open|filtered telnet
25/tcp open|filtered smtp
80/tcp open|filtered http
111/tcp open|filtered rpcbind
139/tcp open|filtered netbios-ssn
445/tcp open|filtered microsoft-ds
512/tcp open|filtered exec
513/tcp open|filtered login
514/tcp open|filtered shell
1099/tcp open|filtered rmiregistry
1524/tcp open|filtered ingreslock
2121/tcp open|filtered ccproxy-ftp
3306/tcp open|filtered mysql
5432/tcp open|filtered postgresql
5900/tcp open|filtered vnc
6000/tcp open|filtered X11
6667/tcp open|filtered irc
8009/tcp open|filtered ajp13
8180/tcp open|filtered
```

Рисунок 40 — Сканирование с использованием флага -sX

После этого было произведено сканирование с помощью флага -sM. Как видно на рисунке 41, при использовании флага -sM найти открытые порты таким способом не удалось.

```
(root@attacker)-[~]
# ./scan.py
Scanning 172.18.0.2 args '-sM' ...

Host: 172.18.0.2 (metasploitable2.pentest) State: up
```

Рисунок 41 — Сканирование с использованием флага -sM

Такой же результат был получен и при использовании флагов -sA и -sW. Это видно на рисунках 42-43.

```
(root@attacker)-[~]
# ./scan.py
Scanning 172.18.0.2 args '-sA' ...

Host: 172.18.0.2 (metasploitable2.pentest) State: up
```

Рисунок 42 — Сканирование с использованием флага -sA

```
└─(root㉿attacker)-[~/]  
└─# ./scan.py  
Scanning 172.18.0.2 args '-sW' ...  
  
Host: 172.18.0.2 (metasploitable2.pentest) State: up
```

Рисунок 43 — Сканирование с использованием флага -sW

Выполнить Idle-сканирование с помощью флага -sI не удалось, поскольку не удалось найти подходящий зомби-хост. Однако можно однозначно утверждать, что результаты скорее всего были бы теми же, что и прежде.

3 Вывод

В ходе выполнения данной лабораторной работы был изучен типовой алгоритм работы с инструментами обнаружения уязвимостей информационных систем, приобретены практические навыки по использованию сканера инфраструктурных уязвимостей, получены навыки идентификации уязвимости информационной системы.