

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

Швалов Даниил Андреевич

**Проверил:**

преподаватель практики, КТН

Назаров Михаил Сергеевич

Санкт-Петербург

2024

## Оглавление

Введение.....	3
Содержание отчета.....	4
1. Установка и проверка корректности работы Docker.....	4
2. Создание лаборатории для тестирования и поиска уязвимостей.....	5
3. Работа со сканером уязвимости OpenVAS.....	9
Вывод по работе.....	14

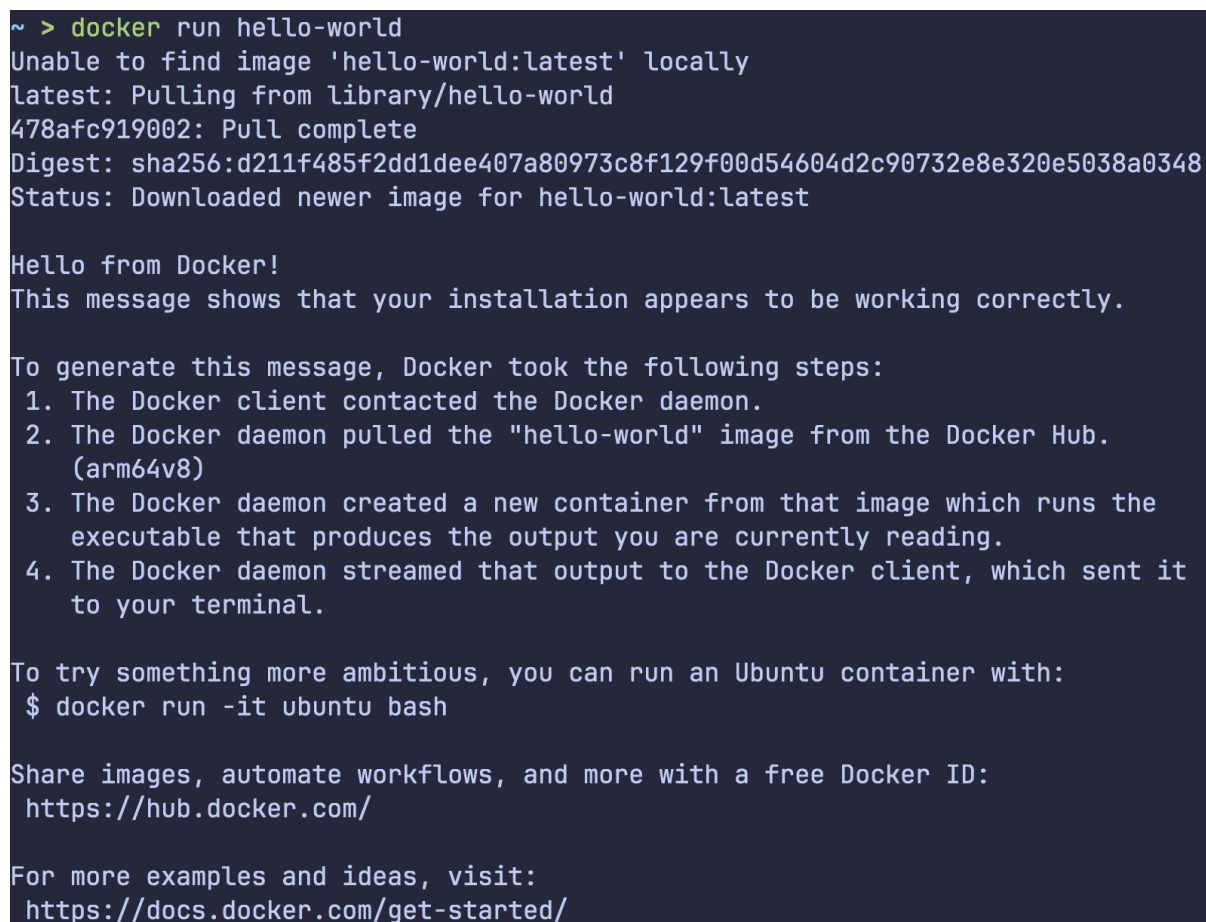
## **Введение**

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

## Содержание отчета

### 1. 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.

A screenshot of a terminal window with a dark background and light-colored text. The text shows the command 'docker run hello-world' being executed. It first reports that the image 'hello-world:latest' was not found locally and is being pulled from the Docker Hub library. It shows the pull progress, including a digest and a status message indicating the image was downloaded. Then, it prints 'Hello from Docker!' followed by a confirmation message that the installation appears to be working correctly. Below this, it lists the steps Docker took to generate the message: 1. The Docker client contacted the Docker daemon. 2. The Docker daemon pulled the 'hello-world' image from the Docker Hub. (arm64v8) 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. Finally, it suggests trying something more ambitious by running an Ubuntu container with the command '\$ docker run -it ubuntu bash'. It also provides a link to share images and automate workflows with a free Docker ID, and a link to visit for more examples and ideas.

```
~ > docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
478afc919002: Pull complete
Digest: sha256:d211f485f2dd1dee407a80973c8f129f00d54604d2c90732e8e320e5038a0348
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.
   (arm64v8)
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»

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.

```
~ > docker run --detach --publish=80:80 --name=webserver nginx
808e9c9b31a3aeeda1103945fcabd634b9f9dc33df4de3732b583b53ecfafabe
```

Рисунок 2 — Запуск контейнера «nginx»

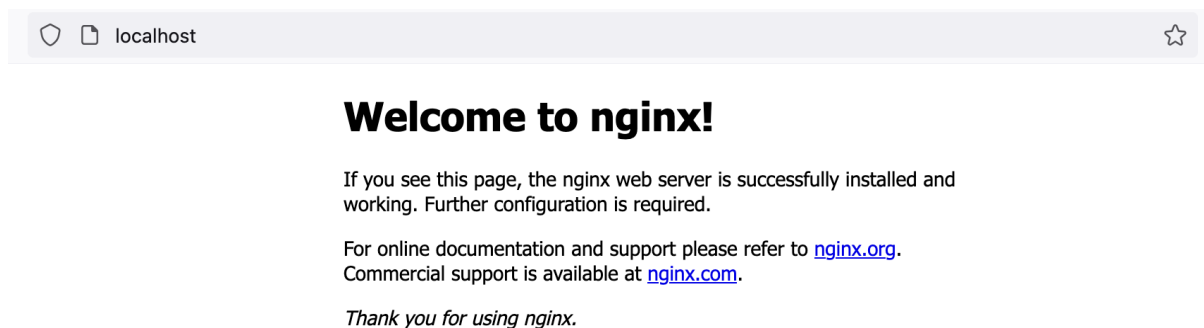


Рисунок 3 — Страница Nginx в браузере

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

```
~ > docker container ls
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS               NAMES
808e9c9b31a3   nginx    "/docker-entrypoint..." 4 minutes ago  Up 4 minutes  0.0.0.0:80->80/tcp   webserver
```

Рисунок 4 — Информация о запущенных контейнерах

## 2. 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.

```

~ > docker pull tleemcjr/metasploitable2
Using default tag: latest
latest: Pulling from tleemcjr/metasploitable2
7aee18c98c59: Pull complete
da9129f8f7ad: Pull complete
b1494b474174: Pull complete
84da87a98ea3: Pull complete
47fb2fcd8445: Pull complete
8b6e3bfdb228: Pull complete
36d703894057: Pull complete
43cf3a9e2a40: Pull complete
Digest: sha256:e559450b37dccc1909eafa2df5b20bb052e1bd801246f4539a3ef183d5f7288a
Status: Downloaded newer image for tleemcjr/metasploitable2:latest
docker.io/tleemcjr/metasploitable2:latest
~ > docker pull kalilinux/kali-rolling
Using default tag: latest
latest: Pulling from kalilinux/kali-rolling
4a6151d0953e: Pull complete
Digest: sha256:8d624392d5b2a9be92f13a579e469eacdde58c2409e78480dd2f6adf628c33c0
Status: Downloaded newer image for kalilinux/kali-rolling:latest
docker.io/kalilinux/kali-rolling:latest

```

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

```

~ > docker images

```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
kalilinux/kali-rolling	latest	d2429a8fce7a	6 days ago	151MB
nginx	latest	12ef77b9fab6	12 months ago	192MB
hello-world	latest	ee301c921b8a	17 months ago	9.14kB
tleemcjr/metasploitable2	latest	db90cb788ea1	6 years ago	1.51GB

Рисунок 6 — Информация о загруженных образах

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

```

~ > docker network create pentest
7d009f9d428ddd5b7b03293295b6fa312cd37ac6a7c05cdf7750121707699d45

```

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.

```

~ > docker run --network=pentest -h victim -it --rm --name metasploitable2 tleemcjr/metasploitable2
WARNING: The requested image's platform (linux/amd64) does not match the detected host platform (linux/arm64/v8) and no specific platform was requested
* 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 [ OK ]
* Starting periodic command scheduler crond [ OK ]
Starting distccd [ OK ]
* Starting MySQL database server mysqld [ OK ]
* Checking for corrupt, not cleanly closed and upgrade needing tables.
* Configuring network interfaces... [ OK ]
* Starting portmap daemon... [ OK ]
* Starting Postfix Mail Transport Agent postfix [ OK ]
* Starting PostgreSQL 8.3 database server [ OK ]
* Starting ftp server proftpd [ OK ]
Starting Samba daemons: nmbd smbdc
Starting network management services: snmpd.
snmpd[1928]: error finding row index in _ifXTable_container_row_restore

* Starting OpenBSD Secure Shell server sshd [ OK ]
* Starting system log daemon... [ OK ]
* Starting Tomcat servlet engine tomcat5.5 [ OK ]
* Starting internet superserver xinetd [ OK ]
* Doing Wacom setup...
cat: */id: No such file or directory [ OK ]

* Running local boot scripts (/etc/rc.local)
nohup: appending output to `nohup.out'
nohup: appending output to `nohup.out' [ OK ]

root@victim:/#

```

Рисунок 8 — Запуск контейнера «metasploitable2»

```

~ > docker run --network=pentest -h attacker -it --rm --name kalibox kalilinux/kali-rolling
#

```

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.

```
(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 2335 bytes 20689198 (19.7 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 2226 bytes 148168 (144.6 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
    loop txqueuelen 1000 (Local Loopback)
    RX packets 8 bytes 646 (646.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 8 bytes 646 (646.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

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

```
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=0.291 ms
64 bytes from 172.18.0.3: icmp_seq=2 ttl=64 time=0.390 ms
64 bytes from 172.18.0.3: icmp_seq=3 ttl=64 time=0.347 ms
64 bytes from 172.18.0.3: icmp_seq=4 ttl=64 time=0.463 ms
64 bytes from 172.18.0.3: icmp_seq=5 ttl=64 time=0.247 ms
64 bytes from 172.18.0.3: icmp_seq=6 ttl=64 time=0.223 ms
64 bytes from 172.18.0.3: icmp_seq=7 ttl=64 time=0.205 ms
64 bytes from 172.18.0.3: icmp_seq=8 ttl=64 time=0.319 ms
^C
--- 172.18.0.3 ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7190ms
rtt min/avg/max/mdev = 0.205/0.310/0.463/0.084 ms
```

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.

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

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.

```
~ > docker run --network=pentest -h attacker -it --rm --publish=9392:9392 --name kalibox kalilinux/kali-rolling
(root@attacker)-[/]
#
```

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.

```
~ > 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
WARNING: The requested image's platform (linux/amd64) does not match the detected host platform (linux/arm64/v8) and no specific platform was requested
8c7e0e724a54e799d6b0ba1b70c5c714dbeacf624c92b29bb027681efe73762e
```

Рисунок 14 — Запуск контейнера «openvas»

### 3. 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.

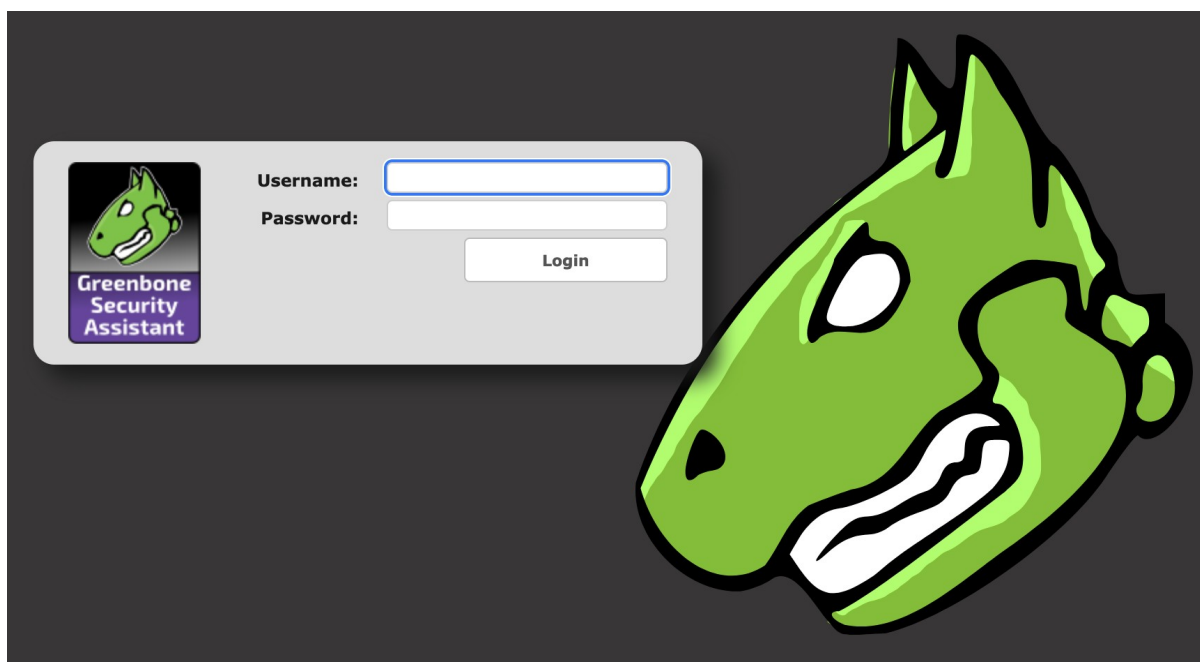


Рисунок 15 — Страница входа в веб-интерфейсе OpenVAS

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.

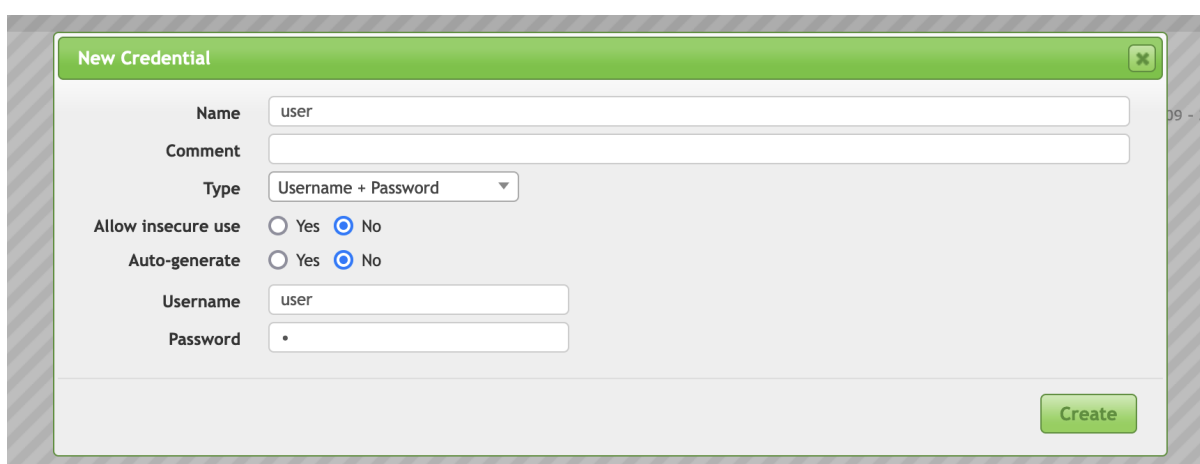


Рисунок 16 — Создание учетной записи в OpenVAS



## Credentials (1 of 1)

Name	Type
user	up (username + password)

(Applied filter: rows=10 first=1 sort=name)

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\_metasploit2», 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

**New Target**

Name: ITMO\_metasploit2

Comment:

Hosts: ☒ Manual 172.18.0.2  
☐ From file Обзор... Файл не выбран.  
☐ From host assets (0 hosts)

Exclude Hosts:

Reverse Lookup Only: ☐ Yes ☒ No

Reverse Lookup Unify: ☐ Yes ☒ No

Port List: All TCP and Nmap 5.51 top... ★

Alive Test: Scan Config Default

Credentials for authenticated checks

SSH: user on port 22 ★

SMB: -- ★

ESXi: -- ★

SNMP: -- ★

Create

Рисунок 18 — Создание цели в OpenVAS



## Targets (1 of 1)

Name	Hosts	IPs
ITMO_metasploit2	172.18.0.2	1

(Applied filter: rows=10 first=1 sort=name)

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\_metasploit2». После создания задачи она появилась в списке «Tasks». Это показано на рисунках 20-21.

**New Task**

Name: SHVALOV\_K34211

Comment:

Scan Targets: ITMO\_metasploit2

Alerts:

Schedule: -- ☐ Once

Add results to Assets: ☒ yes ☐ no

Apply Overrides: ☒ yes ☐ no

Min QoD: 70 %

Alterable Task: ☐ yes ☒ no

Auto Delete Reports: ☒ Do not automatically delete reports  
☐ Automatically delete oldest reports but always keep newest 5 reports

Scanner: OpenVAS Default

Scan Config: Full and fast

Network Source Interface:

Order for target hosts: Sequential

Maximum concurrently executed NVTs per host: 4

Maximum concurrently scanned hosts: 20

Create

Рисунок 20 — Создание задачи в OpenVAS

Name	Status	Reports		Severity	Trend	Actions
		Total	Last			
SHVALOV_K34211	New					

(Applied filter: min\_qod=70 apply\_overrides=1 rows=10 first=1 sort=name)

1 - 1 of 1

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.

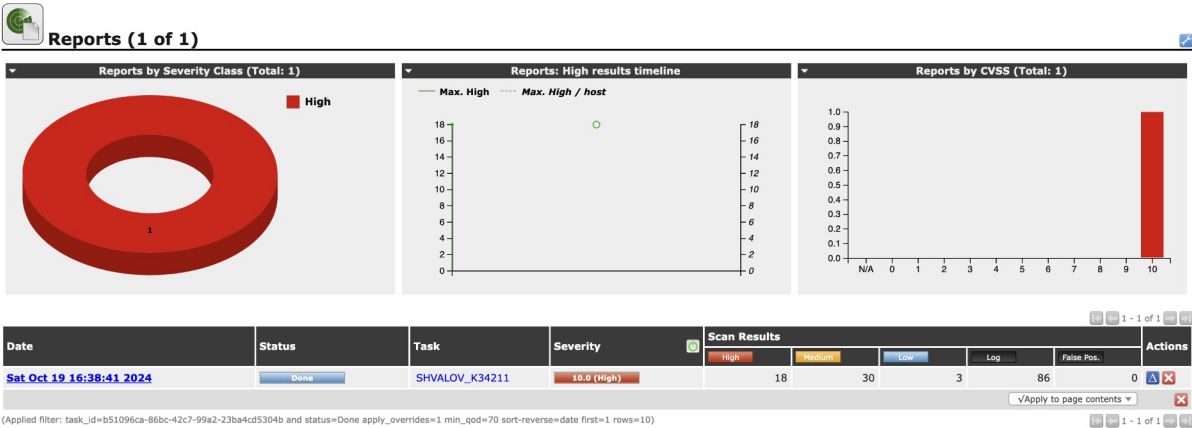



Рисунок 22 — Сформированный отчёт после выполнения задачи



# Report: Results (51 of 361)

ID: 4500a3cd-df51-4696-82e4-754dc5eb1c0d

Modified: Sat Oct 19 17:29:42 2024

Created: Sat Oct 19 16:38:50 2024

Owner: admin

1 - 51 of 361

Actions

Vulnerability	Severity	QoD	Host	Location	Actions
rexec Passwordless / Unencrypted Cleartext Login	10.0 (High)	80%	172.18.0.2 (metasploitable2.pentest)	512/tcp	
OS End Of Life Detection	10.0 (High)	80%	172.18.0.2 (metasploitable2.pentest)	general/tcp	
Twiki XSS and Command Execution Vulnerabilities	10.0 (High)	80%	172.18.0.2 (metasploitable2.pentest)	80/tcp	
Possible Backdoor: Ingreslock	10.0 (High)	99%	172.18.0.2 (metasploitable2.pentest)	1524/tcp	
DistCC Remote Code Execution Vulnerability	9.3 (High)	99%	172.18.0.2 (metasploitable2.pentest)	3632/tcp	
MySQL / MariaDB weak password	9.0 (High)	95%	172.18.0.2 (metasploitable2.pentest)	3306/tcp	
PostgreSQL weak password	9.0 (High)	99%	172.18.0.2 (metasploitable2.pentest)	5432/tcp	
VNC Brute Force Login	9.0 (High)	95%	172.18.0.2 (metasploitable2.pentest)	5900/tcp	
rsh Unencrypted Cleartext Login	7.5 (High)	80%	172.18.0.2 (metasploitable2.pentest)	514/tcp	
phpinfo() output Reporting	7.5 (High)	80%	172.18.0.2 (metasploitable2.pentest)	80/tcp	
Tiki Wiki CMS Groupware < 4.2 Multiple Unspecified Vulnerabilities	7.5 (High)	80%	172.18.0.2 (metasploitable2.pentest)	80/tcp	
PHP-CGI-based setups vulnerability when parsing query string parameters from php files.	7.5 (High)	95%	172.18.0.2 (metasploitable2.pentest)	80/tcp	
Check for Backdoor in UnrealIRCd	7.5 (High)	70%	172.18.0.2 (metasploitable2.pentest)	6697/tcp	
Check for Backdoor in UnrealIRCd	7.5 (High)	70%	172.18.0.2 (metasploitable2.pentest)	6667/tcp	
vstftpd Compromised Source Packages Backdoor Vulnerability	7.5 (High)	99%	172.18.0.2 (metasploitable2.pentest)	6200/tcp	
vstftpd Compromised Source Packages Backdoor Vulnerability	7.5 (High)	99%	172.18.0.2 (metasploitable2.pentest)	21/tcp	
Test HTTP dangerous methods	7.5 (High)	99%	172.18.0.2 (metasploitable2.pentest)	80/tcp	
SSH Brute Force Logins With Default Credentials Reporting	7.5 (High)	95%	172.18.0.2 (metasploitable2.pentest)	22/tcp	
UnrealIRCd Authentication Spoofing Vulnerability	6.8 (Medium)	80%	172.18.0.2 (metasploitable2.pentest)	6697/tcp	
UnrealIRCd Authentication Spoofing Vulnerability	6.8 (Medium)	80%	172.18.0.2 (metasploitable2.pentest)	6667/tcp	
Twiki Cross-Site Request Forgery Vulnerability - Sep10	6.8 (Medium)	80%	172.18.0.2 (metasploitable2.pentest)	80/tcp	

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 информационной системы.