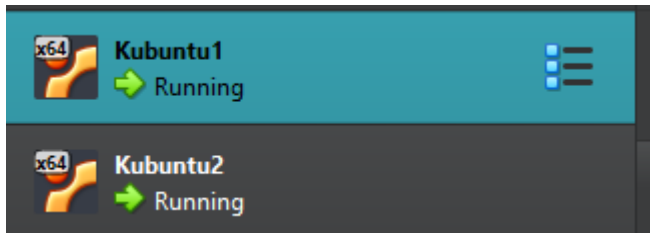


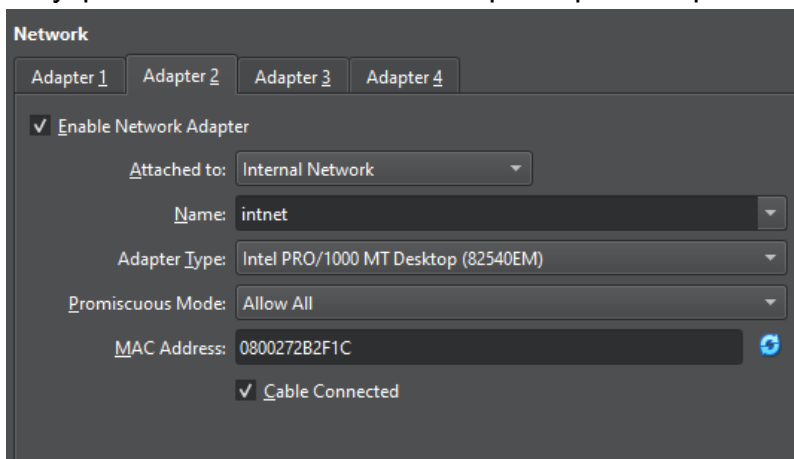
Практическая работа 6

Павлович Джурдже М4105

1. Установить 2 виртуальные машины. Я установил Kubuntu 24.04.01 на Virtualbox.



2. Настроить на них первый сетевой интерфейс - NAT по умолчанию, а второй – типа “внутренняя сеть”. Включите “неразборчивый режим”



3. В самой VM настроить IP адрес VM на 10.10.10.3/29 и 10.10.10.4/29

a:

BM1:

```
djurdje1@kubuntu1:~$ sudo ip addr add 10.10.10.3/29 dev enp0s8
[sudo] password for djurdje1:
djurdje1@kubuntu1:~$ sudo ip link set enp0s8 up
djurdje1@kubuntu1:~$
```

BM2:

```
djurdje2@kubuntu2:~$ sudo ip addr add 10.10.10.4/29 dev enp0s8
[sudo] password for djurdje2:
djurdje2@kubuntu2:~$ sudo ip link set enp0s8 up
djurdje2@kubuntu2:~$
```

- b: *Оформить /etc/network/interfaces,

BM1:

```
#loopback network interface
auto lo
iface lo inet loopback

#NAT network interface
auto enp0s3
iface enp0s3 inet dhcp

#internal network interface
auto enp0s8
iface enp0s8 inet static
    address 10.10.10.3
    netmask 255.255.255.248
    gateway 10.10.10.1
```

BM2:

```
GNU nano 7.2
#loopback network interface
auto lo
iface lo inet loopback

#NAT network interface
auto enp0s3
iface enp0s3 inet dhcp

#internal network interface
auto enp0s8
iface enp0s8 inet static
    address 10.10.10.4
    netmask 255.255.255.248
    gateway 10.10.10.1
```

4. IP v4. Проверьте, что в машине включен форвардинг IP v4 пакетов

Потом нужно сделать перенаправление IPv4 постоянным: там уже есть команда, ее надо только откомментировать.

```
djurdje1@kubuntu1:~$ sysctl net.ipv4.ip_forward
net.ipv4.ip_forward = 0
djurdje1@kubuntu1:~$ sudo sysctl -w net.ipv4.ip_forward=1
net.ipv4.ip_forward = 1
djurdje1@kubuntu1:~$ sudo nano /etc/sysctl.conf
```

```
# Uncomment the next line to enable packet forwarding for IPv4
net.ipv4.ip_forward=1
```

Проверка:

```
djurdje1@kubuntu1:~$ sudo sysctl -p
net.ipv4.ip_forward = 1
djurdje1@kubuntu1:~$
```

5. Установите пакет wireguard

```
9 packages can be upgraded. Run 'apt list --upgradable' to see them.
djurdje2@kubuntu2:~$ sudo apt install wireguard
Reading package lists... Done
```

BM1:

```
djurdje1@kubuntu1:~$ sudo sh -c 'wg genkey | tee /etc/wireguard/privatekey | wg pubkey | tee /etc/wireguard/publickey'
mMPwOYTQALC07dcYkD4hcydad1zJhD2E2DJkpKMMETc=
djurdje1@kubuntu1:~$
```

BM2:

```
djurdje2@kubuntu2:~$ sudo sh -c 'wg genkey | tee /etc/wireguard/privatekey | wg pubkey | tee /etc/wireguard/publickey'
pI9CQjKCDHsA5k16YcxUTwAgUPPr3IfysHnH9t/5DUw=
```

7. Сделайте конфиг в /etc/wireguard/wg0.conf

```
GNU nano 7.2 /etc/wireguard/wg0.conf
[Interface]

Address = 172.16.10.1/29

PostUp = iptables -A FORWARD -i %i -j ACCEPT; iptables -t nat -A POSTROUTING -o enp0s3 -j MASQUERADE
PostDown = iptables -D FORWARD -i %i -j ACCEPT; iptables -t nat -D POSTROUTING -o enp0s3 -j MASQUERADE

ListenPort = 51820

PrivateKey = kG80PKgMlzbKJIoS3H3rvdlQVq84jKXFQ/ttW3+NmDcHK=
```

Во флаге `-o` указывается интерфейс, который будет использоваться для туннеля в настройках WireGuard. Обычно это внешний сетевой интерфейс.

`PostUp` — это набор команд, которые выполняются после того, как интерфейс WireGuard был поднят, а `PostDown` - набор команд, которые выполняются после того опущения интерфейса WireGuard.

`iptables -A FORWARD -i %i -j ACCEPT:`

- Эта команда добавляет правило в цепочку `FORWARD` `iptables`, разрешающее передачу пакетов, поступающих на интерфейс WireGuard (`%i`). Это необходимо для маршрутизации трафика через туннель.

`iptables -t nat -A POSTROUTING -o <INTERFACE> -j MASQUERADE:`

- Эта команда добавляет правило в таблицу NAT (Network Address Translation), которое выполняет **маскарадинг** для исходящих пакетов через указанный внешний интерфейс

1. Запустите туннель командой `wg-quick up wg0`
2. Проверьте статус туннеля командой `wg show wg0`
3. Приложите вывод в отчет

```

[Interface]

Address = 172.16.10.1/29

PostUp = iptables -A FORWARD -i %i -j ACCEPT; iptables -t nat -A POSTROUTING -o enp0s3 -j MASQUERADE
PostDown = iptables -D FORWARD -i %i -j ACCEPT; iptables -t nat -D POSTROUTING -o enp0s3 -j MASQUERADE

ListenPort = 51820

PrivateKey = kg80PKgMlzbKJIOSH3rmdlQVq84jXFQ/ttW3+NmDcHk=

[Peer]
PublicKey = pI9CQjKCDHsA5k16YcxUTwAgUPPr3IfysHnH9t/5DUw=
AllowedIPs = 172.16.10.3/32

```

```

djurdje2@kubuntu2:~$ sudo wg show wg0
interface: wg0
  public key: mMPwOYTQALC07dcYkD4hcydad1zJhD2E2DJkpKMMETc=
  private key: (hidden)
  listening port: 51820

peer: pI9CQjKCDHsA5k16YcxUTwAgUPPr3IfysHnH9t/5DUw=
  endpoint: 10.10.10.4:49600
  allowed ips: 172.16.10.3/32
  latest handshake: 13 seconds ago
  transfer: 7.08 KiB received, 6.96 KiB sent

```

BM2:

```

GNU nano 7.2 /etc/wireguard/wg0.conf
[Interface]

PrivateKey = +MI8hjwa9FKWvMyxlZl30x4IhQCMs0QlvVLC4SpkMGs=

Address = 172.16.10.3/29

PostUp = iptables -A FORWARD -i %i -j ACCEPT; iptables -t nat -A POSTROUTING -o enp0s3 -j MASQUERADE
PostDown = iptables -D FORWARD -i %i -j ACCEPT; iptables -t nat -D POSTROUTING -o enp0s3 -j MASQUERADE

[Peer]

PublicKey = mMPwOYTQALC07dcYkD4hcydad1zJhD2E2DJkpKMMETc=

Endpoint = 10.10.10.3:51820

AllowedIPs = 172.16.10.0/29

PersistentKeepalive = 20

```

```

djurdje2@kubuntu2:~$ sudo wg show wg0
interface: wg0
  public key: pI9CQjKCDHsA5k16YcxUTwAgUPPr3IfysHnH9t/5DUw=
  private key: (hidden)
  listening port: 49600

peer: mMPwOYTQALC07dcYkD4hcydad1zJhD2E2DJkpKMMETc=
  endpoint: 10.10.10.3:51820
  allowed ips: 172.16.10.0/29
  transfer: 0 B received, 296 B sent
  persistent keepalive: every 20 seconds
djurdje2@kubuntu2:~$

```

Проверка на VM1:

```
transfer: 7.00 KB received, 0.90 KB sent
djurdje1@kubuntu1:~$ sudo nano /etc/wireguard/wg0.conf
djurdje1@kubuntu1:~$ ping 172.16.10.3
PING 172.16.10.3 (172.16.10.3) 56(84) bytes of data.
64 bytes from 172.16.10.3: icmp_seq=1 ttl=64 time=0.886 ms
64 bytes from 172.16.10.3: icmp_seq=2 ttl=64 time=0.342 ms
64 bytes from 172.16.10.3: icmp_seq=3 ttl=64 time=0.392 ms
64 bytes from 172.16.10.3: icmp_seq=4 ttl=64 time=0.410 ms
64 bytes from 172.16.10.3: icmp_seq=5 ttl=64 time=0.415 ms
64 bytes from 172.16.10.3: icmp_seq=6 ttl=64 time=0.382 ms
```

```
019CQJKCDHSA5KI6YCX0TWAGUFPF3ItySHH9L/5DOW=
djurdje2@kubuntu2:~$ ping 172.16.10.1
PING 172.16.10.1 (172.16.10.1) 56(84) bytes of data.
64 bytes from 172.16.10.1: icmp_seq=1 ttl=64 time=0.385 ms
64 bytes from 172.16.10.1: icmp_seq=2 ttl=64 time=0.362 ms
64 bytes from 172.16.10.1: icmp_seq=3 ttl=64 time=0.383 ms
64 bytes from 172.16.10.1: icmp_seq=4 ttl=64 time=0.412 ms
64 bytes from 172.16.10.1: icmp_seq=5 ttl=64 time=0.428 ms
64 bytes from 172.16.10.1: icmp_seq=6 ttl=64 time=0.370 ms
```

Проверка на VM2:

15. Поднять на втором сервере базу данных Postgres, сделать так, чтобы она была доступна только по адресу тунеля. Показать, что с первого сервера получится сделать подключение.

Установка PostgreSQL (на VM2): (он установится и запустится автоматически)

```
sudo apt update
sudo apt install postgresql
```

- Откройте файл (путь может отличаться, например у меня postgresql 16 поэтому мой путь - /etc/postgresql/14/main/postgresql.conf).
- Найдите строку `#listen_address` которая закомментированная и измените ее.

```
# - Connection Settings -

#listen_addresses = 'localhost'          # what IP address(es) to listen on;
#                                           # comma-separated list of addresses;
#                                           # defaults to 'localhost'; use '*' for
#                                           # all (only works on Unix-like oses)

listen_addresses = '172.16.10.3'         # comma-separated list of addresses;
```

Сейчас PostgreSQL слушать только на адресе WireGuard.

В файле `/etc/postgresql/16/main/pg_hba.conf` нужно добавить разрешение для сети WireGuard:

```
host      replication  all          *:1/128      scram-sha-256
host      all          all          172.16.10.0/29  md5
```

Также нужно после этого перезапустить PostgreSQL. Теперь PostgreSQL слушает только на 172.16.10.3:5432.

Проверка соединения с первой машины (VM1):

Подключитесь по туннелю:

```
psql -h 172.16.10.3 -U postgres -d postgres
```

```
5
djurdje1@kubuntu1:~$ psql -h 172.16.10.3 -U postgres -d postgres
Password for user postgres:
psql (16.6 (Ubuntu 16.6-0ubuntu0.24.04.1))
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, compression: off)
Type "help" for help.

postgres=#
```

16. Запишите подключение с использованием команды tcpdump, полученный файл откройте в Wireshark. Покажите трафик, который шел через туннель, как выглядят пакеты

На второй машине (VM2), где работает PostgreSQL, запустил команду:

```
sudo tcpdump -i wg0 -w postgres_traffic.pcap
```

Параллельно, на первой машине (VM1) мы подключились к PostgreSQL и выполнили несколько SQL-команд (например, \dt, \t)

```
5
djurdje2@kubuntu2:~$ sudo tcpdump -i wg0 -w postgres_traffic.pcap
tcpdump: listening on wg0, link-type RAW (Raw IP), snapshot length 262144 bytes
^C44 packets captured
44 packets received by filter
0 packets dropped by kernel
djurdje2@kubuntu2:~$
```

После завершения захвата трафика, нужно установить Wireshark.

```
sudo apt install wireshark
```

postgres_traffic.pcap						
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help						
Apply a display filter ... <Ctrl-/>						
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	172.16.10.1	172.16.10.3	TCP	60	52926 → 5432 [SYN]
2	0.000034	172.16.10.3	172.16.10.1	TCP	60	5432 → 52926 [SYN]
3	0.000294	172.16.10.1	172.16.10.3	TCP	52	52926 → 5432 [ACK]
4	0.001120	172.16.10.1	172.16.10.3	PGSQL	60	>?
5	0.001134	172.16.10.3	172.16.10.1	TCP	52	5432 → 52926 [ACK]
6	0.001485	172.16.10.3	172.16.10.1	PGSQL	53	<
7	0.001867	172.16.10.1	172.16.10.3	TCP	52	52926 → 5432 [ACK]
8	0.003623	172.16.10.1	172.16.10.3	TLSv1.3	345	Client Hello
9	0.003890	172.16.10.3	172.16.10.1	TLSv1.3	151	Hello Retry Request
10	0.004312	172.16.10.1	172.16.10.3	TLSv1.3	384	Change Cipher Spec
11	0.005736	172.16.10.3	172.16.10.1	TLSv1.3	1396	Server Hello, Appl
12	0.006877	172.16.10.1	172.16.10.3	TLSv1.3	126	Application Data
13	0.006891	172.16.10.1	172.16.10.3	TLSv1.3	158	Application Data

Frame 1: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface	0000	45 00 00 3c cf 59 40 00	40 06 ff 3d ac 10 0a
Raw packet data	0010	ac 10 0a 03 ce be 15 38	7c 0b 59 08 00 00 06
Internet Protocol Version 4, Src: 172.16.10.1, Destination: 172.16.10.3	0020	a0 02 fd 5c 27 b0 00 00	02 04 05 64 04 02 08
Transmission Control Protocol, Src Port: 52926, Destination Port: 5432	0030	a0 e8 5d 2b 00 00 00 00	01 03 03 07

17. Написать iptables правило, заменяющее DNS Google (8.8.8.8) на DNS Яндекса (77.88.8.8). Найдите способ доказать, что правило работает (подсказка; счетчики)

**На машине (в этом случае VM1) добавляем правило:

```
sudo iptables -t nat -A OUTPUT -p udp -d 8.8.8.8 --dport 53 -j DNAT --to-destination 77.88.8.8`
```

Далее был выполнен DNS-запрос:

```
dig @8.8.8.8 google.com
```

Проверка работы правила iptables через tcpdump:

выполнен захват сетевого трафика с помощью команды:

```
djurdje1@kubuntu1:~$ sudo tcpdump -i enp0s3 host 77.88.8.8
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on enp0s3, link-type EN10MB (Ethernet), snapshot length 262144 bytes
05:34:59.569204 IP kubuntu1.40585 > dns.yandex.ru.domain: 43626+ [1au] A? google.com. (51)
05:34:59.646606 IP dns.yandex.ru.domain > kubuntu1.40585: 43626 1/0/1 A 142.250.74.142 (55)
```

Через эту команду видно что правило iptables успешно перенаправило трафик на 8.8.8.8 к 77.88.8.8, и запросы DNS обрабатывались сервером Яндекса.

17. Напишите iptables правило, запрещающее вашей машине подключаться на сайт itmo.ru. 18. Докажите работу правила

Сначала нужно узнать IP сайта itmo.ru. Можно сделать с командой

```
dig itmo.ru
```

В секции ANSWER указан IP адрес.

```
;; QUESTION SECTION:
;itmo.ru.                IN      A

;; ANSWER SECTION:
itmo.ru.                  4885    IN      A      51.250.120.146
```

Добавление правила iptables:

```
sudo iptables -A OUTPUT -p tcp -d 51.250.120.146 -j REJECT
```

```
IT02::1      tps-attnodes  tps-localhost  kubuntu1
djurdje1@kubuntu1:~$ sudo iptables -A OUTPUT -p tcp -d 51.250.120.146 -j REJECT
[sudo] password for djurdje1:
djurdje1@kubuntu1:~$
```

-j REJECT : Запрещает соединение и отправляет ответ об отказе.

Проверка:

```
djurdje1@kubuntu1:~$ curl itmo.ru
curl: (7) Failed to connect to itmo.ru port 80 after 98 ms: Couldn't connect to server
djurdje1@kubuntu1:~$
```

Unable to connect

An error occurred during a connection to itmo.ru.

- The site could be temporarily unavailable or too busy. Try again in a few moments.
- If you are unable to load any pages, check your computer's network connection.
- If your computer or network is protected by a firewall or proxy, make sure that Firefox is permitted to access the web.

Try Again

На второй машине где не добавил правило можно подключится:

```
Processing triggers for man-db (2.12.1-1ubuntu1) ...  
djurdje2@kubuntu2:~$ curl itmo.ru  
<html>  
<head><title>308 Permanent Redirect</title></head>  
<body>  
<center><h1>308 Permanent Redirect</h1></center>  
<hr><center>nginx</center>  
</body>  
</html>  
djurdje2@kubuntu2:~$
```