

Цель работы:

Закрепить понимание принципов работы NAT и firewall, а также сформировать начальные навыки в конфигурировании NAT и Firewall на платформе и Linux;

Требования:

Установленная на компьютере среда виртуализации ORACLE Virtual Box с виртуальной машиной Linux Cent OS 7 (выполнять работу можно в любой ОС Linux, но все описания будут даваться для CentOS 7).

Инструментальные средства:

Утилиты: sysctl systemctl ip ping tcpdump useradd ss netstat iptables iptables-save iptables-restore

Файлы: /etc/ssh/sshd_config

Утилиты работы с текстом: echo, grep, sed

Редакторы: vi, nano

Порядок выполнения работы:

Примечание: вместо iptables можно выполнить работу на nftables.

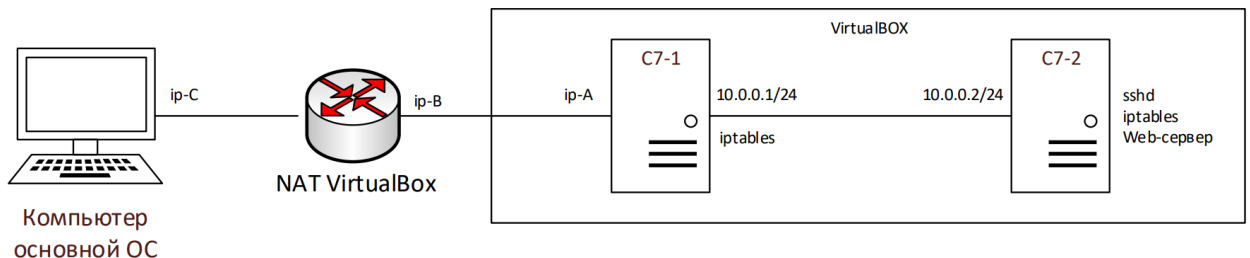
Часть 1. Подготовка и проверка конфигурации

В VirtualBox:

1. Запустите виртуальную машину Linux. Удалите на хосте сервис firewalld.
Примечание: можно использовать утилиту systemctl. Для остановки сервиса используйте команду systemctl stop, для запуска systemctl start, для запрета автозапуска systemctl disable, для включения автозагрузки сервиса systemctl enable.
2. Установите iptables (пакет называется iptables-services), настройте автозапуск iptables.
3. Сделайте связанный клон виртуальной машины. Одну машину назовите c7-1, другой c7-2
4. Для виртуальной машины c7-1 добавьте второй сетевой интерфейс.
5. Подключите сетевой интерфейс c7-2 и новый сетевой интерфейс c7-1 к внутренней сети intnet.

6. Подключите исходный сетевой интерфейс c7-1 к NAT.
7. Для внутренней сети задайте для машин c7-1 и c7-2 адреса 10.0.0.1 и 10.0.0.2 с маской 255.255.255.0.
8. Для исходного интерфейса c7-1 оставьте получение адреса автоматически от dhcp сервера VirtualBox
9. Для обоих хостов отключите использование ipv6.
10. Задайте имена хостов, советуящие именам виртуальных машин. Изменить имя хоста можно изменить с помощью утилиты `hostnamectl`
11. Проверьте доступность хостов по внутренней сети и доступность внешней сети на хосте c7-1.
12. Убедитесь, что на c7-2 в качестве шлюза по умолчанию задан адрес c7-1.
13. В качестве адреса DNS сервера на c7-2 указать адрес 8.8.8.8 и 77.88.8.1
14. Убедитесь, что на машине c7-1 параметры ядра позволяют передавать сетевые пакеты между сетевыми интерфейсами.

Должна получиться следующая схема:



Часть 2. Создание пользователей и настройка OpenSSH Server (sshd).

1. На хосте c7-2 создайте пользователя с именем FIOuser, где FIO – ваши инициалы.
2. Редактируя файл `/etc/ssh/sshd_config`, настройте ssh сервер так, чтобы (!):
 - a. Пользователю root нельзя было бы входить по ssh
 - b. Количество попыток ввода неверного пароля = 2
 - c. Время ожидания авторизации = 30 секундам.
 - d. Отключить определение имен хостов по DNS
3. После изменения конфигурации перезапустите сервис sshd.

4. С машины c7-1 подключитесь к c7-2 по ssh, используя новую учетную запись.

Часть 3. Настройка NAT на шлюзе

1. На хосте c7-1 разрешите передачу IP пакетов между интерфейсами.
2. Настройте на хосте клиентский NAT (действие SNAT или MASQUERADE), так чтобы внешняя сеть стала доступна из внутренней сети.
3. Настройте публикацию порта tcp\22 на хосте c7-2 на порту tcp\55022 на внешнем сетевом интерфейсе c7-1.
4. Используя утилиту iptables-save выведите автоматически созданные правила в текстовый файл /etc/sysconfig/iptables . Определите назначение каждой строки.
5. Подключитесь к ssh серверу на c7-2 с вашей реальной операционной системы (предварительно настройте публикацию портов в NAT в VirtualBox).
6. Проверьте командой ping с хоста c7-2 доступность любого работающего сервиса в Интернет (например адреса 8.8.8.8 или 77.88.8.1). Если хост недоступен, а подключение в п.5 удалось установить, то отредактируйте файл /etc/sysconfig/iptables, изменив правила так, чтобы запросы утилиты ping проходили. Для применения правил можно просто перезапустить сервис (systemctl reload или restart). Корректнее использовать iptables-restore (текущие соединения не сбрасываются).

Часть 4. Установка дополнительного ПО

1. На хосте c7-1 установите консольный броузер (lynx или links) и утилиту nmap.
2. На хосте c7-2 установите Web-сервер lighttpd, запустите его и разрешите автоматический запуск. Определите на каком сокете запускается сервер. Если по умолчанию он стартует на сокете ipv6, то измените конфигурационный файл Web-сервера, так, чтобы сервер запускался на ipv4.
3. С хоста c7-1 с помощью утилиты nmap проверьте какие порты открыты на хосте c7-2 (!).

4. На хосте c7-1 с помощью консольного броузера попробуйте открыть сайт на 10.0.0.2. Если сайт не отрывается, отредактируйте правила iptables, так, чтобы доступ к web-серверу был разрешен. Проверьте, что доступ появился.

Часть 5. Исследование соединений

1. На хосте c7-2 с помощью команд ss, netstat и lsof (любой из команд) выведите на консоль информацию о (!):
 - a. Открытых соединениях.
 - b. Открытых сетевых сокетах, ждущих подключение.
2. На машине c7-1 с помощью утилиты tcpdump выведите на разных консолях трафик с внутреннего и внешнего интерфейса, так чтобы отображались адреса отправителя и получателя, номера портов отправителя и получателя,
3. Запустите с хоста c7-2 передачу 5 TCP сегментов до хоста ya.ru с помощью утилиты mtr.
4. Наблюдая за консольными выводами tcpdump определите, как были изменены исходящие сообщения при трансляции адресов (!).
5. Закройте все ssh сессии с машиной c7-2
6. На машине c7-2 запустите с помощью утилиты tcpdump выведите консоль трафик, так чтобы отображались адреса отправителя и получателя, номера портов отправителя и получателя и флаги tcp (!).
7. Подключитесь с основной операционной системы к хосту c7-2 по ssh.
8. Определите какие флаги использовались при установлении соединения, как менялось значение полей ack и syn после начала передачи данных (!).

Примечание: значения флагов в выводе tcpdump следующие [.] - ACK (Acknowledgment), [S] - SYN (Start Connection); [P] - PSH (Push Data); [F] - FIN (Finish Connection); [R] - RST (Reset Connection); [S.] - SYN-ACK (SynAck Packet)

Часть 6. Настройка шлюза

1. Задайте политики по умолчанию для цепочек INPUT и FORWARD – запрет передачи.
2. Добавьте правила, которые бы

- а. Разрешали подключение к опубликованному порту ssh сервера c7-2 из IP сети реального хоста
- б. Разрешили подключение из внутренней сети к DNS только на 8.8.8.8 и 77.88.8.1
- в. Разрешали доступ из внутренней сети к протоколам POP3 (tcp 110), Web (tcp 80, 443, 8080), ssh (tcp 22).
- г. Разрешили доступ к сервисам SMTP (tcp 25) на любом хосте сети вашего основного компьютера.
- д. Запрещают любой трафик с хостов 192.56.0.11 и с подсети 14.12.44.0/18 как непосредственно на машину c7-1, так и во внутреннюю сеть.
- е. Запрещают доступ к ssh серверу на c7-1 из внешней сети.
- ж. Разрешает доступ к ssh серверу на c7-1 из внутренней сети.
- з. Разрешает icmp эхо запросы из внутренней сети наружу только на хост 8.8.8.8
- и. Запрещает хосту c7-1 давать icmp эхо ответы, но при этом сохраняет возможность с самого хоста c7-1 делать icmp это запросы и получать на них ответы.

Часть 7. Доступ через ssh к защищенным сервисам

Используя возможности протокола ssh сделайте так, чтобы на основном компьютере Web-сервер с хоста c7-2 был доступен по адресу 127.0.0.80:8888.

Артефакты:

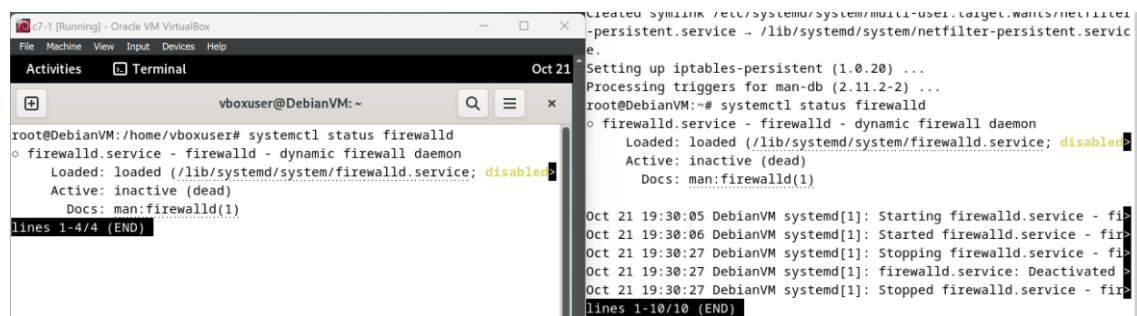
1. Измененные параметры sshd из Части 2.

Часть №1

Отключили на c7-1 и c7-2 firewalld:

```
systemctl stop firewalld
```

```
systemctl disable firewalld
```



```
root@DebianVM: /home/vboxuser# systemctl status firewalld
○ firewalld.service - firewalld - dynamic firewall daemon
  Loaded: loaded (/lib/systemd/system/firewalld.service; disabled; vendor preset: enabled)
  Active: inactive (dead)
  Docs: man:firewalld(1)
lines 1-4/4 (END)

Created symlink /etc/systemd/system/multi-user.target.wants/netfilter-persistent.service → /lib/systemd/system/netfilter-persistent.service.
Setting up iptables-persistent (1.0.20) ...
Processing triggers for man-db (2.11.2-2) ...
root@DebianVM:~# systemctl status firewalld
○ firewalld.service - firewalld - dynamic firewall daemon
  Loaded: loaded (/lib/systemd/system/firewalld.service; disabled; vendor preset: enabled)
  Active: inactive (dead)
  Docs: man:firewalld(1)
lines 1-10/10 (END)

Oct 21 19:30:05 DebianVM systemd[1]: Starting firewalld.service - firewalld - dynamic firewall daemon
Oct 21 19:30:06 DebianVM systemd[1]: Started firewalld.service - firewalld - dynamic firewall daemon
Oct 21 19:30:27 DebianVM systemd[1]: Stopping firewalld.service - firewalld - dynamic firewall daemon
Oct 21 19:30:27 DebianVM systemd[1]: firewalld.service: Deactivated successfully.
Oct 21 19:30:27 DebianVM systemd[1]: Stopped firewalld.service - firewalld - dynamic firewall daemon
lines 1-10/10 (END)
```

Установили на c7-1 и c7-2 iptables и настроили автозапуск:

```
apt install iptables iptables-persistent -y
```

```
systemctl enable netfilter-persistent
```

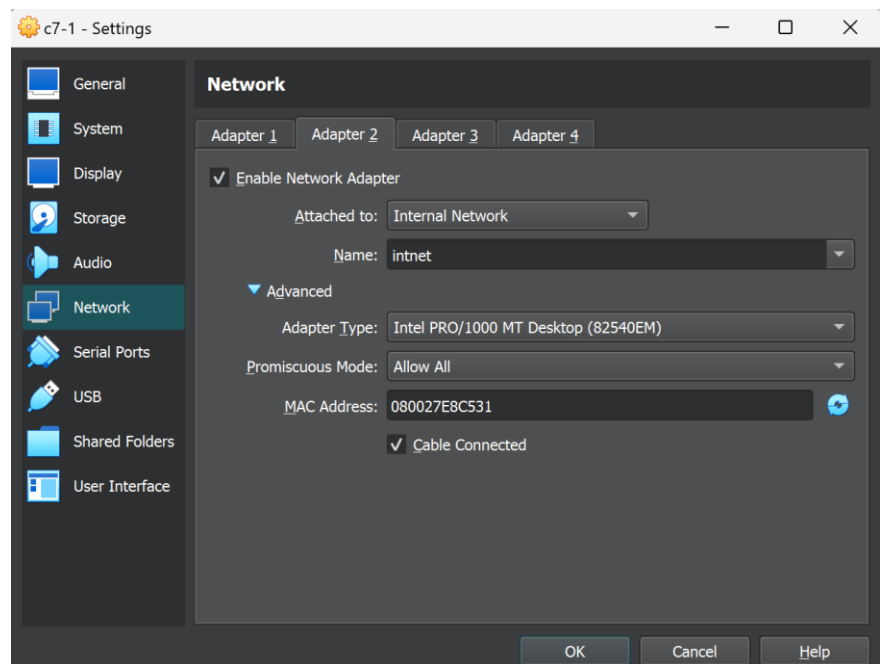
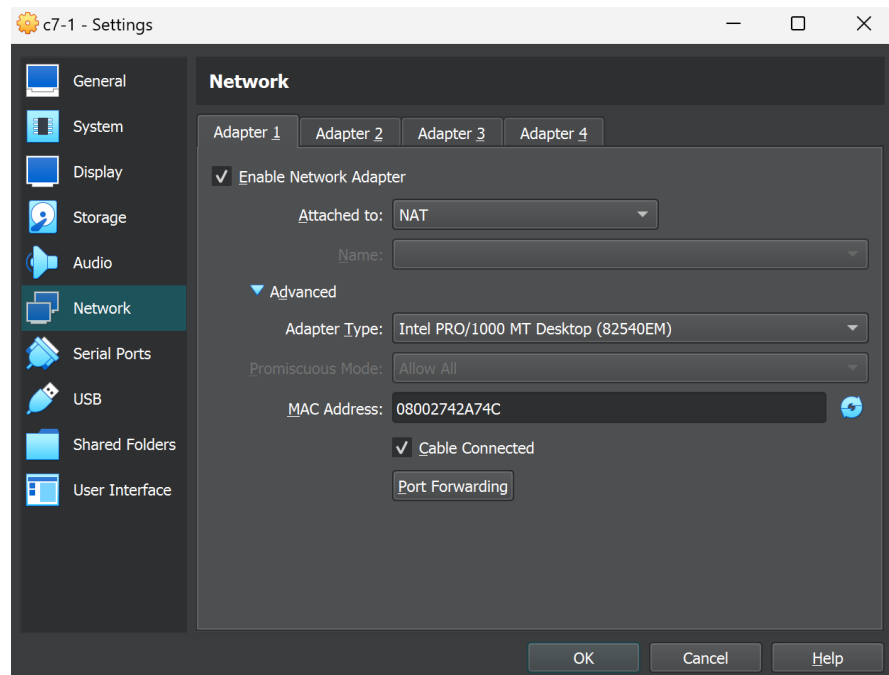
```
systemctl start netfilter-persistent
```

```
root@DebianVM:~# systemctl enable netfilter-persistent
Synchronizing state of netfilter-persistent.service with SysV service script with
/usr/sbin/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable netfilter-persistent
root@DebianVM:~# systemctl start netfilter-persistent
root@DebianVM:~# systemctl status netfilter-persistent
● netfilter-persistent.service - netfilter persistent configuration
  Loaded: loaded (/lib/systemd/system/netfilter-persistent.service; enabled; vendor preset: enabled)
  Drop-In: /usr/lib/systemd/system/netfilter-persistent.service.d
           └─iptables.conf
  Active: active (exited) since Mon 2024-10-21 19:39:54 MSK; 55s ago
  Docs: man:netfilter-persistent(8)
  Main PID: 2328 (code=exited, status=0/SUCCESS)
  CPU: 20ms

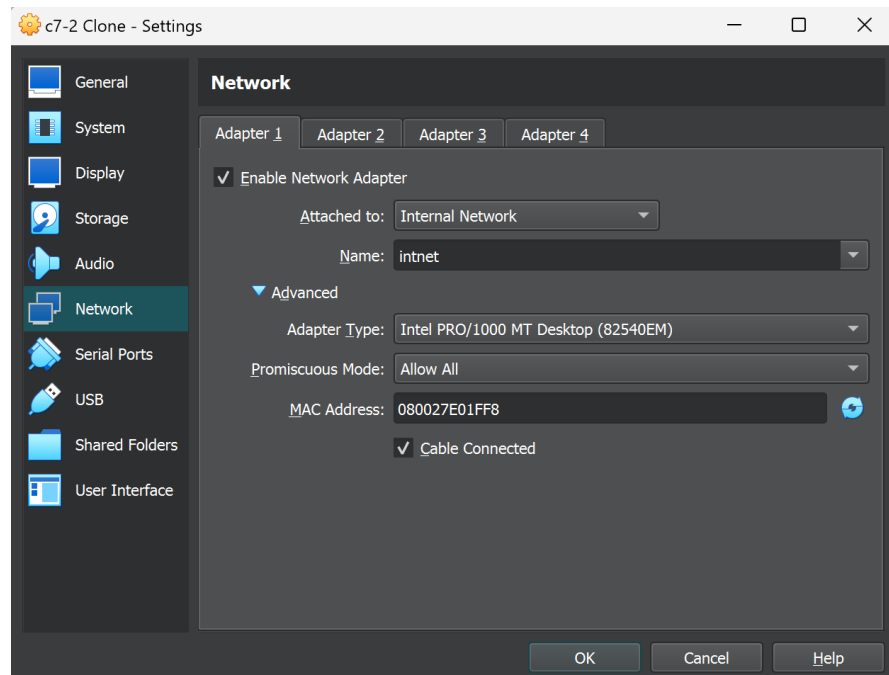
Oct 21 19:39:54 DebianVM systemd[1]: Starting netfilter-persistent.service - netfilter persistent configuration
Oct 21 19:39:54 DebianVM netfilter-persistent[2330]: run-parts: executing /usr/sbin/iptables
Oct 21 19:39:54 DebianVM netfilter-persistent[2331]: Warning: skipping IPv4 (no rules)
Oct 21 19:39:54 DebianVM netfilter-persistent[2330]: run-parts: executing /usr/sbin/iptables
Oct 21 19:39:54 DebianVM netfilter-persistent[2332]: Warning: skipping IPv6 (no rules)
Oct 21 19:39:54 DebianVM netfilter-persistent[2332]: /usr/share/netfilter-persistent/scripts/iptables
Oct 21 19:39:54 DebianVM netfilter-persistent[2332]: Error: IPv6 rules failed to save
Oct 21 19:39:54 DebianVM systemd[1]: Finished netfilter-persistent.service - netfilter persistent configuration
lines 1-17/17 (END)
```

Настройка сетевых интерфейсов для c7-1 и c7-2:

Для c7-1:



Для c7-2:



Для внутренней сети зададим для машин c7-1 и c7-2 адреса 10.0.0.1 и 10.0.0.2:

c7-1

```
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP
    link/ether 08:00:27:42:a7:4c brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic enp0s3
        valid_lft 558sec preferred_lft 558sec

3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP
    link/ether 08:00:27:e8:c5:31 brd ff:ff:ff:ff:ff:ff
    inet 10.0.0.1/24 scope global enp0s8
        valid_lft forever preferred_lft forever
```

c7-2

```
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP
    link/ether 08:00:27:e0:1f:f8 brd ff:ff:ff:ff:ff:ff
    inet 10.0.0.2/24 scope global enp0s3
        valid_lft forever preferred_lft forever
```

Отключили IPv6 на обеих машинах:

```
sudo sysctl -w net.ipv6.conf.all.disable_ipv6=1
```

```
sudo sysctl -w net.ipv6.conf.default.disable_ipv6=1
```

Изменение имени хоста:

C7-1:

```
sudo hostnamectl set-hostname c7-1
```

C7-2:

```
sudo hostnamectl set-hostname c7-2
```

Проверка доступности по внутренней сети:

```
vboxuser@DebianVM:~/labs/lab1$ ping 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=0.784 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.749 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.949 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=1.06 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=1.69 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=1.18 ms
^C
--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5008ms
rtt min/avg/max/mdev = 0.749/1.069/1.687/0.314 ms

vboxuser@DebianVM:~$ ping 10.0.0.1
PING 10.0.0.1 (10.0.0.1) 56(84) bytes of data.
64 bytes from 10.0.0.1: icmp_seq=1 ttl=64 time=0.863 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=0.865 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=0.880 ms
64 bytes from 10.0.0.1: icmp_seq=4 ttl=64 time=1.05 ms
64 bytes from 10.0.0.1: icmp_seq=5 ttl=64 time=0.951 ms
64 bytes from 10.0.0.1: icmp_seq=6 ttl=64 time=1.02 ms
64 bytes from 10.0.0.1: icmp_seq=7 ttl=64 time=0.692 ms
64 bytes from 10.0.0.1: icmp_seq=8 ttl=64 time=0.889 ms
^C
--- 10.0.0.1 ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7009ms
rtt min/avg/max/mdev = 0.692/0.901/1.047/0.103 ms
```

Настройка шлюза по умолчанию на c7-2:

```
sudo ip route add default via 10.0.0.1
```

Настройка DNS серверов на c7-2:

```
sudo nano /etc/resolv.conf
```

```
nameserver 8.8.8.8
```

```
nameserver 77.88.8.1
```

Разрешение передачи пакетов между интерфейсами на c7-1:

```
sudo sysctl -w net.ipv4.ip_forward=1
```

Часть 2

Создание пользователя на c7-2:

```
sudo useradd FIOuser
```

```
sudo passwd FIOuser (пароль 239)
```

Настройка SSH на c7-2:

```
sudo nano /etc/ssh/sshd_config
```

Внесли изменения:

Запрет для root:

`PermitRootLogin no`

Количество попыток ввода пароля:

`MaxAuthTries 2`

Время ожидания авторизации:

`LoginGraceTime 30`

Отключение определения имен хостов по DNS:

`UseDNS no`

Перезапуск службы SSH: (ДОДЕЛАТЬ)

`sudo systemctl restart sshd`

Подключение по SSH с c7-1 на c7-2:

`ssh FIUser@10.0.0.2`

`root@c7-1: /home/vboxuser/labs/lab1# ssh FIUser@10.0.0.2`

█

2. Итоговые файлы /etc/sysconfig/iptables с хостов c7-1 и c7-2

Часть №3

Разрешим пересылку пакетов на c7-1:

`sudo sysctl -w net.ipv4.ip_forward=1`

Настройка SNAT/MASQUERADE на c7-1:

`sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE`

Публикация порта SSH:

Настройка переадресации порта:

`sudo iptables -t nat -A PREROUTING -p tcp --dport 55022 -j`

`DNAT --to-destination 10.0.0.2:22`

Сохранение iptables в файл:

`sudo iptables-save > /home/vboxuser/labs/sixlabrestable`

```

GNU nano 7.2 /home/vboxuser/labs/sixlabrestable
# Generated by iptables-save v1.8.9 (nf_tables) on Wed Oct 30 19:39:05 2024
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A FORWARD -d 10.0.0.2/32 -p tcp -m tcp --dport 22 -j ACCEPT
COMMIT
# Completed on Wed Oct 30 19:39:05 2024
# Generated by iptables-save v1.8.9 (nf_tables) on Wed Oct 30 19:39:05 2024
*nat
:PREROUTING ACCEPT [0:0]
:INPUT ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
:POSTROUTING ACCEPT [0:0]
-A PREROUTING -p tcp -m tcp --dport 55022 -j DNAT --to-destination 10.0.0.2:22
-A POSTROUTING -o eth0 -j MASQUERADE
-A POSTROUTING -o enp0s8 -j MASQUERADE
COMMIT
# Completed on Wed Oct 30 19:39:05 2024

```

Подключение к c7-2 через c7-1 по SSH с реальной ОС:

```
ssh -p 55022 FIOuser@localhost
```

Проверка доступа в Интернет:

```
ping 8.8.8.8
```

```

PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=58 time=24.4 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=58 time=8.92 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=58 time=12.1 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=58 time=267 ms
^C
--- 8.8.8.8 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3020ms
rtt min/avg/max/mdev = 8.916/78.098/267.025/109.229 ms

```

3. Команду и консольный вывод из Части 4 п.3

Часть №4

Установка lynx и nmap на c7-1:

```
sudo apt install lynx nmap
```

Установка Web-сервера на c7-2:

```
sudo apt install lighttpd
```

Запуск и включение автозапуска:

```
sudo systemctl start lighttpd
```

```
sudo systemctl enable lighttpd
```

```

root@c7-2:/home/vboxuser/labs/lab1# sudo systemctl status lighttpd
● lighttpd.service - Lighttpd Daemon
   Loaded: loaded (/lib/systemd/system/lighttpd.service; enabled; preset: enabled)
   Active: active (running) since Wed 2024-10-30 20:03:32 MSK; 1min 43s ago
     Main PID: 2762 (lighttpd)
        Tasks: 1 (limit: 2293)
       Memory: 900.0K
          CPU: 348ms
      CGroup: /system.slice/lighttpd.service
              └─2762 /usr/sbin/lighttpd -D -f /etc/lighttpd/lighttpd.conf

Oct 30 20:03:32 c7-2 systemd[1]: Starting lighttpd.service - Lighttpd Daemon...
Oct 30 20:03:32 c7-2 systemd[1]: Started lighttpd.service - Lighttpd Daemon.

```

Сканирование портов с помощью nmap на c7-1:

```
sudo nmap 10.0.0.2
```

```

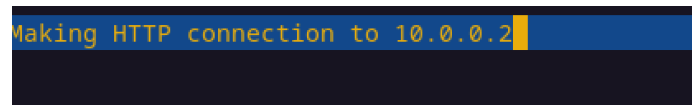
root@c7-1:~# nmap 10.0.0.2
Starting Nmap 7.93 ( https://nmap.org ) at 2024-10-30 20:06 MSK
Nmap scan report for 10.0.0.2
Host is up (0.00049s latency).
All 1000 scanned ports on 10.0.0.2 are in ignored states.
Not shown: 1000 filtered tcp ports (no-response)

Nmap done: 1 IP address (1 host up) scanned in 5.47 seconds

```

Открытие сайта на c7-2 с c7-1:

```
lynx 10.0.0.2
```



4. Команды и существенные части консольного вывода Части 5, п. 1,4,6,8

Часть №5

Вывод информации о соединениях на c7-2:

Открытые соединения:

```

root@c7-2:/home/vboxuser/labs/lab1# ss -tuna

```

Netid	State	Recv-Q	Send-Q	Local Address:Port	Peer Address:Port	Process
udp	UNCONN	0	0	0.0.0.0:68	0.0.0.0:*	
udp	ESTAB	0	0	10.0.3.15%enp0s8:68	10.0.3.2:67	
udp	UNCONN	0	0	0.0.0.0:5353	0.0.0.0:*	
udp	UNCONN	0	0	0.0.0.0:57824	0.0.0.0:*	
udp	UNCONN	0	0	:::5353	:::*	
udp	UNCONN	0	0	:::59018	:::*	
tcp	LISTEN	0	1024	0.0.0.0:80	0.0.0.0:*	
tcp	LISTEN	0	128	0.0.0.0:22	0.0.0.0:*	
tcp	LISTEN	0	128	127.0.0.1:631	0.0.0.0:*	
tcp	LISTEN	0	128	:::1:631	:::*	
tcp	LISTEN	0	1024	:::80	:::*	
tcp	LISTEN	0	128	:::22	:::*	

```

root@c7-2:/home/vboxuser/labs/lab1#

```

Открытые сетевые сокеты, ожидающие подключений:

```

root@c7-2:/home/vboxuser/labs/lab1# netstat -ltnp
Active Internet connections (only servers)

```

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State	PID/Program name
tcp	0	0	0.0.0.0:80	0.0.0.0:*	LISTEN	2762/lighttpd
tcp	0	0	0.0.0.0:22	0.0.0.0:*	LISTEN	659/sshd: /usr/sbin
tcp	0	0	127.0.0.1:631	0.0.0.0:*	LISTEN	638/cupsd
tcp6	0	0	:::1:631	:::*	LISTEN	638/cupsd
tcp6	0	0	:::80	:::*	LISTEN	2762/lighttpd
tcp6	0	0	:::22	:::*	LISTEN	659/sshd: /usr/sbin

Дополнительная информация о сокетах:

```
root@c7-2: /home/vboxuser/labs/lab1# lsof -i
COMMAND  PID  USER  FD  TYPE  DEVICE  SIZE/OFF  NODE NAME
avahi-daemon 403  avahi 12u  IPv4  15739      0t0  UDP *:mdns
avahi-daemon 403  avahi 13u  IPv6  15740      0t0  UDP *:mdns
avahi-daemon 403  avahi 14u  IPv4  15741      0t0  UDP *:57824
avahi-daemon 403  avahi 15u  IPv6  15742      0t0  UDP *:59018
NetworkManager 459  root 29u  IPv4  16168      0t0  UDP c7-2:bootpc->_gateway:bootps
cupsd 638  root 6u  IPv6  16154      0t0  TCP localhost:ipp (LISTEN)
cupsd 638  root 7u  IPv4  16155      0t0  TCP localhost:ipp (LISTEN)
sshd 659  root 3u  IPv4  17277      0t0  TCP *:ssh (LISTEN)
sshd 659  root 4u  IPv6  17288      0t0  TCP *:ssh (LISTEN)
dhclient 2226  root 8u  IPv4  23003      0t0  UDP *:bootpc
lighttpd 2762  www-data 4u  IPv6  29790      0t0  TCP *:http (LISTEN)
lighttpd 2762  www-data 5u  IPv4  29791      0t0  TCP *:http (LISTEN)
```

Перехват трафика с помощью tcpdump на c7-1:

Открыли 2 терминала на c7-1.

В первом терминале начните прослушивание трафика на внешнем интерфейсе:

```
sudo tcpdump -i enp0s3 -nn
```

```
root@c7-1:~# sudo tcpdump -i enp0s3 -nn
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on enp0s3, link-type EN10MB (Ethernet), snapshot length 262144 bytes
20:14:58.779529 IP 10.0.2.15 > 8.8.8.8: ICMP echo request, id 41409, seq 1, length 64
20:14:58.800772 IP 8.8.8.8 > 10.0.2.15: ICMP echo reply, id 41409, seq 1, length 64
20:14:59.780386 IP 10.0.2.15 > 8.8.8.8: ICMP echo request, id 41409, seq 2, length 64
20:14:59.791956 IP 8.8.8.8 > 10.0.2.15: ICMP echo reply, id 41409, seq 2, length 64
20:15:00.785156 IP 10.0.2.15 > 8.8.8.8: ICMP echo request, id 41409, seq 3, length 64
20:15:00.801396 IP 8.8.8.8 > 10.0.2.15: ICMP echo reply, id 41409, seq 3, length 64
20:15:01.787541 IP 10.0.2.15 > 8.8.8.8: ICMP echo request, id 41409, seq 4, length 64
20:15:01.801576 IP 8.8.8.8 > 10.0.2.15: ICMP echo reply, id 41409, seq 4, length 64
20:15:04.000617 ARP, Request who-has 10.0.2.1 tell 10.0.2.15, length 28
20:15:04.000647 ARP, Reply 10.0.2.1 is-at 52:54:00:12:35:00, length 46
```

Во втором терминале прослушивайте трафик на внутреннем интерфейсе:

```
sudo tcpdump -i enp0s8 -nn
```

```
root@c7-1:/home/vboxuser# sudo tcpdump -i enp0s8 -nn
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on enp0s8, link-type EN10MB (Ethernet), snapshot length 262144 bytes
20:16:28.168173 IP 0.0.0.0.68 > 255.255.255.255.67: BOOTP/DHCP, Request from 08:00:27:e0:1f:f8, length 300
20:16:31.996810 IP6 :: > ff02::16: HBH ICMP6, multicast listener report v2, 2 group record(s), length 48
20:16:32.011428 IP 0.0.0.0.68 > 255.255.255.255.67: BOOTP/DHCP, Request from 08:00:27:e0:c5:31, length 288
20:16:32.042899 IP6 :: > ff02::1:ffe8:c531: ICMP6, neighbor solicitation, who has fe80::a00:27ff:fe8:c531, length 32
20:16:32.042999 IP6 :: > ff02::16: HBH ICMP6, multicast listener report v2, 2 group record(s), length 48
20:16:32.416590 IP6 :: > ff02::16: HBH ICMP6, multicast listener report v2, 2 group record(s), length 48
20:16:33.053246 IP6 fe80::a00:27ff:fe8:c531 > ff02::16: HBH ICMP6, multicast listener report v2, 2 group record(s), length 48
20:16:33.072285 IP6 fe80::a00:27ff:fe8:c531 > ff02::2: ICMP6, router solicitation, length 8
20:16:33.116552 IP6 fe80::a00:27ff:fe8:c531.5353 > ff02::fb.5353: 0 [2q] PTR (QM)? _ipps._tcp.local. PTR (QM)? _ipp._tcp.local. (45)
20:16:33.204584 IP6 fe80::a00:27ff:fe8:c531 > ff02::16: HBH ICMP6, multicast listener report v2, 2 group record(s), length 48
20:16:33.399664 IP 0.0.0.0.68 > 255.255.255.255.67: BOOTP/DHCP, Request from 08:00:27:e0:1f:f8, length 300
20:16:33.994822 IP6 fe80::a00:27ff:fe8:c531.5353 > ff02::fb.5353: 0*- [0q] 2/0/0 (Cache flush) PTR c7-1.local., (Cache flush) AAAA fe80::a00:27ff:fe8:c531 (136)
20:16:34.020807 IP 0.0.0.0.68 > 255.255.255.255.67: BOOTP/DHCP, Request from 08:00:27:e0:c5:31, length 288
20:16:35.121337 IP6 fe80::a00:27ff:fe8:c531.5353 > ff02::fb.5353: 0 [2q] PTR (QM)? _ipps._tcp.local. PTR (QM)? _ipp._tcp.local. (45)
20:16:36.102245 IP6 fe80::a00:27ff:fe8:c531.5353 > ff02::fb.5353: 0*- [0q] 2/0/0 (Cache flush) PTR c7-1.local., (Cache flush) AAAA fe80::a00:27ff:fe8:c531 (136)
20:16:36.637206 IP 0.0.0.0.68 > 255.255.255.255.67: BOOTP/DHCP, Request from 08:00:27:e0:c5:31, length 288
```

Отправка TCP пакетов с c7-2 на ya.ru:

```
mtr -c 5 ya.ru
```

В выводах tcpdump наблюдаем, как изменяется IP-адрес отправителя при пересылке через NAT-интерфейс на c7-1:

```

root@c7-1:/home/vboxuser# sudo tcpdump -i enp0s8 -nn
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on enp0s8, link-type EN10MB (Ethernet), snapshot length 262144 bytes
23:53:51.152023 IP6 fe80::a00:27ff:fee8:c531 > ff02::2: ICMP6, router solicitation, length 8
23:53:51.207232 IP6 fe80::a00:27ff:fee8:c531.5353 > ff02::fb.5353: 0 [2q] PTR (QM)? _ipps._tcp.local. PTR (QM)? _ipp._tcp.local. (45)
23:53:53.805739 IP 0.0.0.0.68 > 255.255.255.255.67: BOOTP/DHCP, Request from 08:00:27:e8:c5:31, length 288
23:54:05.000824 IP6 :: > ff02::16: HBH ICMP6, multicast listener report v2, 3 group record(s), length 68
23:54:06.013389 IP6 :: > ff02::16: HBH ICMP6, multicast listener report v2, 3 group record(s), length 68
23:59:05.090399 IP 0.0.0.0.68 > 255.255.255.255.67: BOOTP/DHCP, Request from 08:00:27:e8:c5:31, length 288
23:59:05.100881 IP6 :: > ff02::16: HBH ICMP6, multicast listener report v2, 2 group record(s), length 48
23:59:05.254714 IP6 :: > ff02::16: HBH ICMP6, multicast listener report v2, 2 group record(s), length 48
23:59:05.406096 IP6 :: > ff02::1:ffe8:c531: ICMP6, neighbor solicitation, who has fe80::a00:27ff:fee8:c531, length 32
23:59:06.429540 IP6 fe80::a00:27ff:fee8:c531 > ff02::16: HBH ICMP6, multicast listener report v2, 2 group record(s), length 48
23:59:06.700120 IP6 fe80::a00:27ff:fee8:c531 > ff02::2: ICMP6, router solicitation, length 8
23:59:07.036983 IP6 fe80::a00:27ff:fee8:c531 > ff02::16: HBH ICMP6, multicast listener report v2, 2 group record(s), length 48
23:59:07.069930 IP6 fe80::a00:27ff:fee8:c531.5353 > ff02::fb.5353: 0*- [0q] 2/0/0 (Cache flush) PTR c7-1.local., (Cache flush) AAAA fe80::a0
0:27ff:fee8:c531 (136)
23:59:07.092836 IP 0.0.0.0.68 > 255.255.255.255.67: BOOTP/DHCP, Request from 08:00:27:e8:c5:31, length 288
23:59:08.250518 IP6 fe80::a00:27ff:fee8:c531.5353 > ff02::fb.5353: 0 [2q] PTR (QM)? _ipps._tcp.local. PTR (QM)? _ipp._tcp.local. (45)
23:59:09.182567 IP6 fe80::a00:27ff:fee8:c531.5353 > ff02::fb.5353: 0*- [0q] 2/0/0 (Cache flush) PTR c7-1.local., (Cache flush) AAAA fe80::a0
0:27ff:fee8:c531 (136)
23:59:09.901060 IP 0.0.0.0.68 > 255.255.255.255.67: BOOTP/DHCP, Request from 08:00:27:e8:c5:31, length 288
23:59:10.485281 IP6 fe80::a00:27ff:fee8:c531 > ff02::2: ICMP6, router solicitation, length 8
23:59:12.367912 IP6 fe80::a00:27ff:fee8:c531.5353 > ff02::fb.5353: 0 [2q] PTR (QM)? _ipps._tcp.local. PTR (QM)? _ipp._tcp.local. (45)
23:59:14.714779 IP 0.0.0.0.68 > 255.255.255.255.67: BOOTP/DHCP, Request from 08:00:27:e8:c5:31, length 288
23:59:18.195896 IP6 fe80::a00:27ff:fee8:c531 > ff02::2: ICMP6, router solicitation, length 8
23:59:20.360019 IP6 fe80::a00:27ff:fee8:c531.5353 > ff02::fb.5353: 0 [2q] PTR (QM)? _ipps._tcp.local. PTR (QM)? _ipp._tcp.local. (45)
23:59:23.077695 IP 0.0.0.0.68 > 255.255.255.255.67: BOOTP/DHCP, Request from 08:00:27:e8:c5:31, length 288
23:59:33.843057 IP6 fe80::a00:27ff:fee8:c531 > ff02::2: ICMP6, router solicitation, length 8
23:59:36.380886 IP6 fe80::a00:27ff:fee8:c531.5353 > ff02::fb.5353: 0 [2q] PTR (QM)? _ipps._tcp.local. PTR (QM)? _ipp._tcp.local. (45)

```

Закрытие сессий SSH и перехват трафика:

Закрываем все SSH-сессии с c7-2:

```
sudo pkill sshd
```

На c7-2 запускаем tcpdump для перехвата всех TCP-пакетов и их флагов:

```
sudo tcpdump -i enp0s8 -nn -v 'tcp[tcpflags] & (tcp-syn|tcp-ack|tcp-fin|tcp-rst) != 0'
```

```

root@c7-2:/home/vboxuser/labs/lab1# tcpdump -i enp0s3 -nn -v 'tcp[tcpflags] & (tcp-syn|tcp-ack|tcp-fin|tcp-rst) != 0'
tcpdump: listening on enp0s3, link-type EN10MB (Ethernet), snapshot length 262144 bytes
00:22:58.135301 IP (tos 0x10, ttl 64, id 8669, offset 0, flags [DF], proto TCP (6), length 60)
  10.0.0.1.37762 > 10.0.0.2.55022: Flags [S], cksum 0xf2a0 (correct), seq 460041906, win 64240, options [mss 1460,sackOK,TS val 63245848 e
  cr 0,nop,wscale 7], length 0
00:22:58.135350 IP (tos 0x10, ttl 64, id 0, offset 0, flags [DF], proto TCP (6), length 40)
  10.0.0.2.55022 > 10.0.0.1.37762: Flags [R.], cksum 0x673e (correct), seq 0, ack 460041907, win 0, length 0
00:23:02.763630 IP (tos 0x10, ttl 64, id 9816, offset 0, flags [DF], proto TCP (6), length 60)
  10.0.0.1.37770 > 10.0.0.2.55022: Flags [S], cksum 0x12f4 (correct), seq 2537160820, win 64240, options [mss 1460,sackOK,TS val 63250476
  ecr 0,nop,wscale 7], length 0
00:23:02.763673 IP (tos 0x10, ttl 64, id 0, offset 0, flags [DF], proto TCP (6), length 40)
  10.0.0.2.55022 > 10.0.0.1.37770: Flags [R.], cksum 0x99a5 (correct), seq 0, ack 2537160821, win 0, length 0

```

Во время подключения на c7-2 в терминале с tcpdump можно увидеть последовательность установления TCP-соединения:

[S] (SYN) — начало подключения.

[S.] (SYN-ACK) — ответ на SYN.

[.] (ACK) подтверждение установления соединения.

Анализ изменения флагов ACK и SYN:

При передаче данных, вы заметили, что:

ACK - флаг подтверждает успешную передачу пакетов,

SYN и SYN-ACK происходят только при установлении соединения,

FIN и RST будут видны при завершении или сбросе соединения.

5. Текст итоговых правил iptables с с7-1.

Часть №6

На с7-1 установим политику запрета для цепочек INPUT и FORWARD:

```
sudo iptables -P INPUT DROP
```

```
sudo iptables -P FORWARD DROP
```

Добавление правил:

Разрешить подключение к SSH на с7-2 (порт 55022) из сети хоста:

```
sudo iptables -A INPUT -p tcp --dport 55022 -s <real_host_network>  
-j ACCEPT
```

Разрешить подключение к DNS (8.8.8.8 и 77.88.8.1) из внутренней сети:

```
sudo iptables -A FORWARD -s 10.0.0.0/24 -d 8.8.8.8 -p udp --dport 53  
-j ACCEPT
```

```
sudo iptables -A FORWARD -s 10.0.0.0/24 -d 77.88.8.1 -p udp --dport  
53 -j ACCEPT
```

Разрешить доступ из внутренней сети к веб-протоколам и POP3, SSH:

```
sudo iptables -A FORWARD -s 10.0.0.0/24 -p tcp -m multiport --  
dports 22,80,443,8080,110 -j ACCEPT
```

Разрешить доступ к SMTP из основной сети:

```
sudo iptables -A FORWARD -p tcp --dport 25 -s <real_host_network>  
-j ACCEPT
```

Запретить весь трафик с 192.56.0.11 и сети 14.12.44.0/18:

```
sudo iptables -A INPUT -s 192.56.0.11 -j DROP
```

```
sudo iptables -A INPUT -s 14.12.44.0/18 -j DROP
```

```
sudo iptables -A FORWARD -s 192.56.0.11 -j DROP
```

```
sudo iptables -A FORWARD -s 14.12.44.0/18 -j DROP
```

Запретить доступ к SSH на с7-1 извне:


```
sudo iptables -A INPUT -p tcp --dport 22 -i enp0s3 -j DROP
```

Разрешить доступ к SSH на c7-1 из внутренней сети:

```
sudo iptables -A INPUT -p tcp --dport 22 -i enp0s8 -j ACCEPT
```

Разрешить ICMP-запросы из внутренней сети наружу на 8.8.8.8:

```
sudo iptables -A FORWARD -s 10.0.0.0/24 -d 8.8.8.8 -p icmp --icmp-type echo-request -j ACCEPT
```

Запретить c7-1 давать ICMP-ответы, но оставить возможность отправлять запросы:

```
sudo iptables -A INPUT -p icmp --icmp-type echo-request -j DROP
```

6. Команду подключения из Части 7, п.1.

Часть №7

Настройка проброса порта для доступа к веб-серверу c7-2:

создания туннеля:

```
ssh -L 127.0.0.80:8888:10.0.0.2:80 FIUser@<c7-1_external_ip> -p 55022
```

После этого можно будет обратиться к адресу 127.0.0.80:8888 в браузере, чтобы открыть веб-страницу на c7-2.

Вопросы и задания:

1. В чем разница между действиями SNAT или MASQUERADE? Когда уместно использовать одно, а когда другое?

Ответ:

- SNAT (Source NAT) — это метод изменения исходного IP-адреса пакетов, отправляемых с локальной сети на внешнюю. SNAT используется, когда известен IP-адрес, который будет использоваться для подмены (например, статический IP). Он позволяет сохранить состояние соединений, что может быть полезно для определенных приложений.
- MASQUERADE — это более динамичный вариант SNAT, используемый обычно для подключения к интернету через динамический IP-адрес. При использовании MASQUERADE Linux автоматически подставляет текущий внешний IP-адрес, что делает

его удобным для случаев, когда IP может изменяться (например, при подключении через DSL).

Когда использовать:

- Используйте SNAT, когда у вас есть фиксированный IP-адрес и вы хотите управлять правилами NAT более точно.
- Используйте MASQUERADE, когда у вас динамический IP, и вам не нужно беспокоиться о его изменениях.

2. Какие цепочки и какие таблицы существуют в iptables по умолчанию?

Ответ: В iptables по умолчанию существуют следующие таблицы:

- filter — основная таблица для управления доступом к трафику (принимает, отклоняет, пересылает).
- nat — используется для NAT (изменение адресов в заголовках пакетов).
- mangle — для изменения заголовков пакетов (например, для QoS).
- raw — для управления необработанными пакетами (например, для исключения от состояния соединений).

По умолчанию в таблице filter существуют следующие цепочки:

- INPUT — для входящих пакетов.
- FORWARD — для пакетов, которые пересылаются через маршрутизатор.
- OUTPUT — для исходящих пакетов.

3. Как добавить новую цепочку? Как перенаправить в нее трафик?

Ответ:

1. Чтобы добавить новую цепочку в iptables:

```
iptables -N <имя_цепочки>
```

2. Чтобы перенаправить трафик в новую цепочку, используйте команду для добавления правила в существующую цепочку (например, INPUT, FORWARD или OUTPUT):

```
iptables -A <существующая_цепочка> -j <имя_цепочки>
```

4. Имеет ли смысл порядок правил?

Ответ: да, порядок правил имеет большое значение в iptables. Правила обрабатываются последовательно, и как только пакет соответствует одному из

правил, дальнейшая проверка прекращается. Это означает, что более специфичные правила должны быть выше в списке, чтобы их могли применять до того, как пакет дойдет до более общих правил.

5. Как с помощью iptables можно реализовать настройки, при которых брандмауэр пропускает пакеты тех соединений, которые были инициированы изнутри. Учтите, что правило позволяло установить соединение, т. е. передать пакеты наружу, так и получать ответы, то есть принять ответные пакеты.

Ответ: для реализации этой настройки вы можете использовать состояние соединения в iptables. Вот пример правил, которые позволяют пропускать пакеты для установленных соединений и вновь создаваемые изнутри:

```
# Разрешить входящие пакеты для установленных соединений  
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
```

```
# Разрешить исходящие соединения
```

```
iptables -A OUTPUT -m state --state NEW,ESTABLISHED -j ACCEPT
```

- Первое правило позволяет принимать входящие пакеты, которые являются частью уже установленных соединений или связаны с ними.
- Второе правило разрешает новым соединениям исходить из внутренней сети и также позволяет принимать ответные пакеты для уже установленных соединений.

Эти правила обеспечат необходимую функциональность для брандмауэра.