МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования
Санкт-Петербургский национальный исследовательский университет ИТМО
Мегафакультет трансляционных информационных технологий

Факультет информационных технологий и программирования

Практическая работа №6. Трансляция адресов в ОС Linux

По дисциплине «Телекоммуникационные системы и технологии»

Выполнил:
студент группы №M3306
Тимофеев Вячеслав

Проверил:
Самигуллин [redacted]

УНИВЕРСИТЕТ ИТМО

Санкт-Петербург
2025

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

Артефакты выполнения:

1. Настройка OpenSSH Server:

Измененный конфиг /etc/ssh/sshd_config со следующими настройками:

a. Пользователю root нельзя было бы входить по ssh: PermitRootLogin no

b. Количество попыток ввода неверного пароля = 2: MaxAuthTries 2

c. Время ожидания авторизации = 30 секундам: LoginGraceTime 30

d. Отключить определение имен хостов по DNS: UseDNS no

2. Настройка NAT на шлюзе:

Вывод cat /etc/sysconfig/iptables с хоста c7-1

```
*filter
:INPUT ACCEPT [0:0]
:FORWARD DROP [0:0]
:OUTPUT ACCEPT [0:0]
-A FORWARD -i enp0s3 -o enp0s8 -m state --state RELATED,ESTABLISHED -j ACCEPT
-A FORWARD -i enp0s8 -o enp0s3 -j ACCEPT
-A FORWARD -p tcp -d 10.0.0.2 --dport 22 -j ACCEPT
COMMIT
# Completed on Sun Mar 31 01:58:31 2024
# Generated by iptables-save v1.4.21 on Sun Mar 31 01:58:31 2024
*nat
:PREROUTING ACCEPT [133:8840]
:INPUT ACCEPT [5:340]
:OUTPUT ACCEPT [30:8297]
:POSTROUTING ACCEPT [30:8297]
-A POSTROUTING -o enp0s3 -j MASQUERADE
-A PREROUTING -p tcp --dport 55022 -j DNAT --to-destination 10.0.0.2:22
COMMIT
```

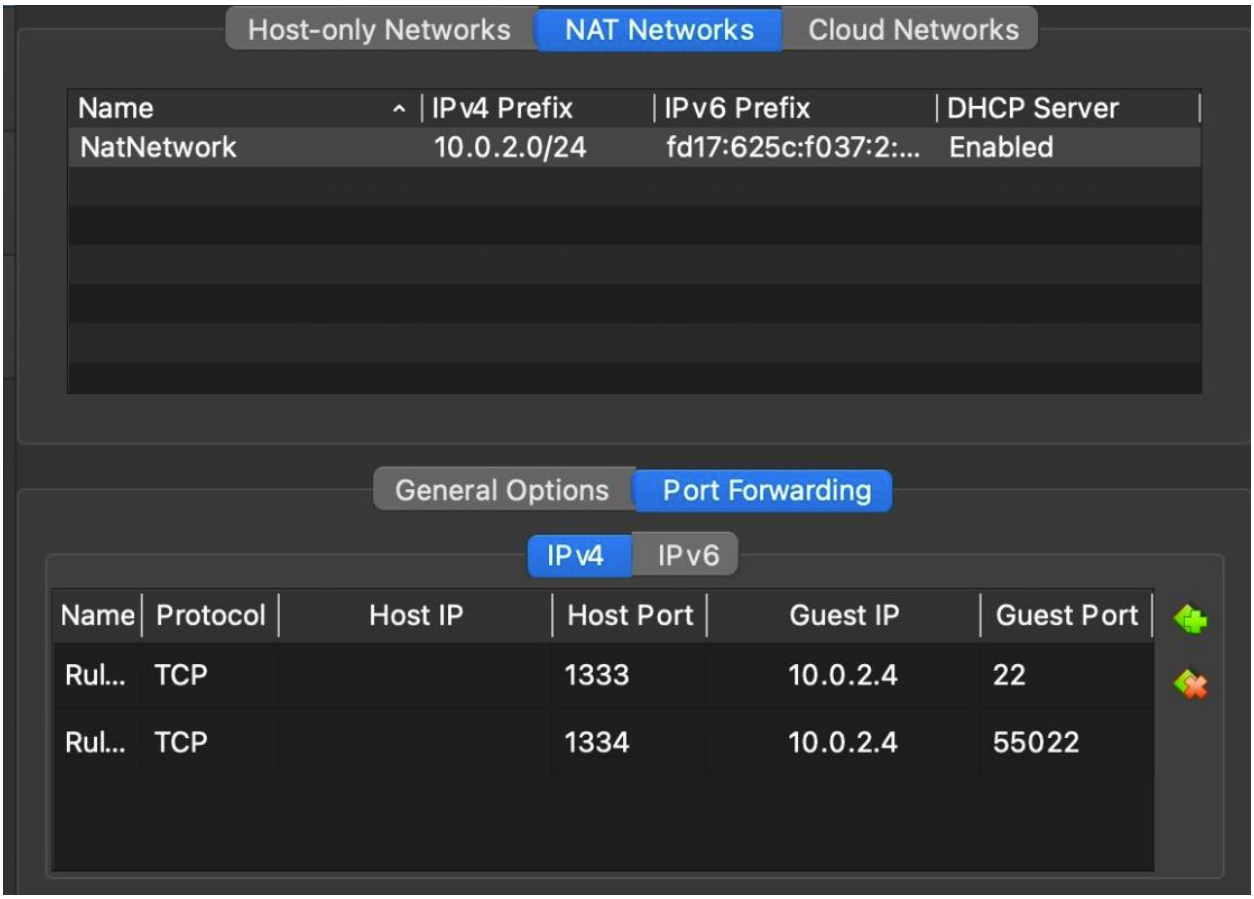
Вывод cat /etc/sysconfig/iptables с хоста c7-2

```
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [6:504]
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 22 -j ACCEPT
COMMIT
```

Подключаемся по ssh:

```
[root@c7-1 ~]# netstat -tulpn | grep LISTEN
tcp        0      0 127.0.0.1:25            0.0.0.0:*               LISTEN      1048/master
tcp        0      0 0.0.0.0:55022           0.0.0.0:*               LISTEN      2187/sshd
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      2187/sshd
```

```
ssh -g -L 55022:10.0.0.2:22 KEGuser@10.0.0.2
```



ssh -p 1334 KEGuser@localhost – команда для подключения

И пингуем

```
[root@c7-2 ~]# ping -c 3 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=108 time=8.83 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=108 time=9.68 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=108 time=7.28 ms

--- 8.8.8.8 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2005ms
rtt min/avg/max/mdev = 7.280/8.601/9.689/1.000 ms
```

4. Установка дополнительного ПО

Проверка открытых портов на c7-2 (nmap -v -St 10.0.0.2)

```
Initiating ARP Ping Scan at 03:27
Scanning 10.0.0.2 [1 port]
Completed ARP Ping Scan at 03:27, 0.20s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 03:27
Completed Parallel DNS resolution of 1 host. at 03:27, 0.02s elapsed
Initiating Connect Scan at 03:27
Scanning 10.0.0.2 [1000 ports]
Discovered open port 22/tcp on 10.0.0.2
Discovered open port 80/tcp on 10.0.0.2
Completed Connect Scan at 03:27, 0.38s elapsed (1000 total ports)
Nmap scan report for 10.0.0.2
Host is up (0.0068s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http
MAC Address: 08:00:27:EB:70:87 (Cadmus Computer Systems)

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 0.65 seconds
           Raw packets sent: 1 (28B) | Rcvd: 1 (28B)
```

Проверяем, что Web-сервер запустился на ipv4 (`netstat -tulpn`)

```
[root@c7-2 ~]# netstat -tulpn
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      1194/lighttpd
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      786/sshd
tcp        0      0 127.0.0.1:25            0.0.0.0:*               LISTEN      1009/master
```

```
Quit                                                    Powered by lighttpd
  LIGHTTPD - fly light.

  Powered by GNU/Linux
  Powered by Lighttpd

Commands: Use arrow keys to move, '?' for help, 'q' to quit, '<-' to go back.
  Arrow keys: Up and Down to move.  Right to follow a link: Left to go back.
H)elp O)ptions P)rint G)o M)ain screen Q)uit /=search [delete]=history list
```

5. Исследование соединений

Вывод информации об открытых соединениях и сетевых сокетах, ждущих подключение

(`netstat -tanp`):

```
[root@c7-2 ~]# netstat -tanp
Active Internet connections (servers and established)
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      1194/lighttpd
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      786/sshd
tcp        0      0 127.0.0.1:25            0.0.0.0:*               LISTEN      1009/master
tcp        0      0 10.0.0.2:80             10.0.0.1:40400          TIME_WAIT   -
tcp        0      0 10.0.0.2:22             10.0.2.2:65298          ESTABLISHED 1637/sshd: KEGuser
```

```
[root@c7-2 ~]# netstat -lx
Active UNIX domain sockets (only servers)
Proto RefCnt Flags       Type       State         I-Node   Path
unix  2      [ ACC ]     STREAM     LISTENING     7441     /run/systemd/journal/stdout
unix  2      [ ACC ]     STREAM     LISTENING     15385    /var/run/NetworkManager/private-dhcp
unix  2      [ ACC ]     STREAM     LISTENING     17031    private/lmtp
unix  2      [ ACC ]     STREAM     LISTENING     17034    private/anvil
unix  2      [ ACC ]     STREAM     LISTENING     17037    private/scache
unix  2      [ ACC ]     STREAM     LISTENING     16979    private/tlsmgr
unix  2      [ ACC ]     STREAM     LISTENING     16982    private/rewrite
unix  2      [ ACC ]     STREAM     LISTENING     16985    private/bounce
unix  2      [ ACC ]     STREAM     LISTENING     16988    private/defer
unix  2      [ ACC ]     STREAM     LISTENING     16991    private/trace
unix  2      [ ACC ]     STREAM     LISTENING     16994    private/verify
unix  2      [ ACC ]     STREAM     LISTENING     17000    private/proxymap
unix  2      [ ACC ]     STREAM     LISTENING     17004    private/proxywrite
unix  2      [ ACC ]     STREAM     LISTENING     16968    public/pickup
unix  2      [ ACC ]     STREAM     LISTENING     17007    private/smtp
unix  2      [ ACC ]     STREAM     LISTENING     17010    private/relay
unix  2      [ ACC ]     STREAM     LISTENING     17016    private/error
unix  2      [ ACC ]     STREAM     LISTENING     17019    private/retry
unix  2      [ ACC ]     STREAM     LISTENING     16972    public/cleanup
unix  2      [ ACC ]     STREAM     LISTENING     16975    public/qmgr
unix  2      [ ACC ]     STREAM     LISTENING     16997    public/flush
unix  2      [ ACC ]     STREAM     LISTENING     17022    private/discard
unix  2      [ ACC ]     STREAM     LISTENING     17025    private/local
unix  2      [ ACC ]     STREAM     LISTENING     17028    private/virtual
unix  2      [ ACC ]     STREAM     LISTENING     13759    /run/dbus/system_bus_socket
unix  2      [ ACC ]     STREAM     LISTENING     11719    /run/systemd/private
unix  2      [ ACC ]     STREAM     LISTENING     17013    public/showq
unix  2      [ ACC ]     SEQPACKET  LISTENING     11744    /run/udev/control
```

Трассировка пакетов на внутренней сети

```
06:21:52.367329 IP (tos 0x0, ttl 64, id 11588, offset 0, flags [DF], proto UDP (17), length 55)
    10.0.0.2.59466 > 8.8.8.8.53: [udp sum ok] 37595+ A? www.ya.ru. (27)
06:21:52.367822 IP (tos 0x0, ttl 64, id 11590, offset 0, flags [DF], proto UDP (17), length 55)
    10.0.0.2.59466 > 8.8.8.8.53: [udp sum ok] 61967+ AAAA? www.ya.ru. (27)
06:21:52.375265 IP (tos 0x0, ttl 254, id 21728, offset 0, flags [none], proto UDP (17), length 101)
    8.8.8.8.53 > 10.0.0.2.59466: [udp sum ok] 37595 q: A? www.ya.ru. 3/0/0 www.ya.ru. [1m40s] CNAME ya.ru., ya.ru. [1m40s] A
77.88.55.242, ya.ru. [1m40s] A 5.255.255.242 (73)
06:21:52.375522 IP (tos 0x0, ttl 254, id 21729, offset 0, flags [none], proto UDP (17), length 97)
    8.8.8.8.53 > 10.0.0.2.59466: [udp sum ok] 61967 q: AAAA? www.ya.ru. 2/0/0 www.ya.ru. [1m40s] CNAME ya.ru., ya.ru. [1m40s] AAAA
2a02:6b8::2:242 (69)
06:21:52.479601 IP (tos 0x0, ttl 1, id 15840, offset 0, flags [DF], proto TCP (6), length 60)
    10.0.0.2.49748 > 77.88.55.242.80: Flags [S], cksum 0x9113 (correct), seq 2955156127, win 29200, options [mss 1460,sackOK,TS
val 11085695 ecr 0,nop,wscale 7], length 0
```

Трассировка пакетов на внешней сети

```
06:21:52.367382 IP (tos 0x0, ttl 63, id 11588, offset 0, flags [DF], proto UDP (17), length 55)
    10.0.2.4.59466 > 8.8.8.8.53: [udp sum ok] 37595+ A? www.ya.ru. (27)
06:21:52.367832 IP (tos 0x0, ttl 63, id 11590, offset 0, flags [DF], proto UDP (17), length 55)
    10.0.2.4.59466 > 8.8.8.8.53: [udp sum ok] 61967+ AAAA? www.ya.ru. (27)
06:21:52.375256 IP (tos 0x0, ttl 255, id 21728, offset 0, flags [none], proto UDP (17), length 101)
    8.8.8.8.53 > 10.0.2.4.59466: [udp sum ok] 37595 q: A? www.ya.ru. 3/0/0 www.ya.ru. [1m40s] CNAME ya.ru., ya.ru. [1m40s] A
77.88.55.242, ya.ru. [1m40s] A 5.255.255.242 (73)
06:21:52.375505 IP (tos 0x0, ttl 255, id 21729, offset 0, flags [none], proto UDP (17), length 97)
    8.8.8.8.53 > 10.0.2.4.59466: [udp sum ok] 61967 q: AAAA? www.ya.ru. 2/0/0 www.ya.ru. [1m40s] CNAME ya.ru., ya.ru. [1m40s] AAAA
2a02:6b8::2:242 (69)
06:21:52.582542 IP (tos 0x0, ttl 1, id 22990, offset 0, flags [DF], proto TCP (6), length 60)
    10.0.2.4.44107 > 77.88.55.242.80: Flags [S], cksum 0x3928 (correct), seq 3089399339, win 29200, options [mss 1460,sackOK,TS
val 11085797 ecr 0,nop,wscale 7], length 0
```

```
06:30:00.449272 IP (tos 0x0, ttl 254, id 22113, offset 0, flags [none], proto TCP (6), length 44)
    10.0.2.2.50979 > 10.0.0.2.22: Flags [S], cksum 0x6e19 (correct), seq 1428666, win 32768, options [mss 1460], length 0
06:30:00.449329 IP (tos 0x0, ttl 64, id 0, offset 0, flags [DF], proto TCP (6), length 44)
    10.0.0.2.22 > 10.0.2.2.50979: Flags [S.], cksum 0x1622 (incorrect -> 0x1fe7), seq 1703868034, ack 1428667, win 29200, options
[mss 1460], length 0
06:30:00.450261 IP (tos 0x0, ttl 254, id 22114, offset 0, flags [none], proto TCP (6), length 40)
    10.0.2.2.50979 > 10.0.0.2.22: Flags [.], cksum 0x29b4 (correct), seq 1, ack 1, win 32768, length 0
06:30:00.450580 IP (tos 0x0, ttl 254, id 22115, offset 0, flags [none], proto TCP (6), length 61)
    10.0.2.2.50979 > 10.0.0.2.22: Flags [P.], cksum 0x67ee (correct), seq 1:22, ack 1, win 32768, length 21
06:30:00.450600 IP (tos 0x0, ttl 64, id 23737, offset 0, flags [DF], proto TCP (6), length 40)
    10.0.0.2.22 > 10.0.2.2.50979: Flags [.], cksum 0x161e (incorrect -> 0x378f), seq 1, ack 22, win 29200, length 0
06:30:00.462501 IP (tos 0x0, ttl 64, id 23738, offset 0, flags [DF], proto TCP (6), length 61)
    10.0.0.2.22 > 10.0.2.2.50979: Flags [P.], cksum 0x1633 (incorrect -> 0x73c9), seq 1:22, ack 22, win 29200, length 21
06:30:00.464139 IP (tos 0x0, ttl 254, id 22116, offset 0, flags [none], proto TCP (6), length 1408)
    10.0.2.2.50979 > 10.0.0.2.22: Flags [P.], cksum 0xece0 (correct), seq 22:1390, ack 22, win 32747, length 1368
06:30:00.466560 IP (tos 0x0, ttl 64, id 23739, offset 0, flags [DF], proto TCP (6), length 1320)
    10.0.0.2.22 > 10.0.2.2.50979: Flags [P.], cksum 0x1b1e (incorrect -> 0xf052), seq 22:1302, ack 1390, win 31464, length 1280
06:30:00.471198 IP (tos 0x0, ttl 254, id 22117, offset 0, flags [none], proto TCP (6), length 88)

    10.0.0.2.22 > 10.0.2.2.50979: Flags [P.], cksum 0x16a2 (incorrect -> 0x3d3e), seq 2578:2710, ack 2770, win 39672, length 132
06:30:04.865301 IP (tos 0x10, ttl 64, id 23754, offset 0, flags [DF], proto TCP (6), length 124)
    10.0.0.2.22 > 10.0.2.2.50979: Flags [P.], cksum 0x1672 (incorrect -> 0x6cae), seq 2710:2794, ack 2770, win 39672, length 84
06:30:04.866502 IP (tos 0x0, ttl 254, id 22128, offset 0, flags [none], proto TCP (6), length 40)
    10.0.2.2.50979 > 10.0.0.2.22: Flags [.], cksum 0x18aa (correct), seq 2770, ack 2794, win 31568, length 0
06:30:05.431707 ARP, Ethernet (len 6), IPv4 (len 4), Request who-has 10.0.0.2 tell 10.0.0.1, length 46
06:30:05.431737 ARP, Ethernet (len 6), IPv4 (len 4), Reply 10.0.0.2 is-at 08:00:27:eb:70:87, length 28
```

6. Настройка шлюза

1. Задайте политики по умолчанию для цепочек INPUT и FORWARD – запрет передачи.

iptables -P INPUT DROP

iptables -P FORWARD DROP

2. Добавьте правила, которые бы

a. Разрешали подключение к опубликованному порту ssh сервера c7-2 из IP сети реального хоста

iptables -A FORWARD -p tcp -d 10.0.0.2 –dport 22 -s 10.0.2.0/24 -j ACCEPT

iptables -A FORWARD -p tcp -s 10.0.0.2 –sport 22 -d 10.0.2.0/24 -j ACCEPT

b. Разрешили подключение из внутренней сети к DNS только на 8.8.8.8 и 77.88.8.1

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

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

iptables -A FORWARD -p udp --sport 53 -s 8.8.8.8 -d 10.0.0.0/24 -j ACCEPT

iptables -A FORWARD -p udp --sport 53 -s 77.88.8.1 -d 10.0.0.0/24 -j ACCEPT

c. Разрешали доступ из внутренней сети к протоколам POP3 (tcp 110), Web (tcp 80, 443, 8080), ssh (tcp 22)

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

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

d. Разрешили доступ к сервисам SMTP (tcp 25) на любом хосте сети вашего основного компьютера.

```
iptables -A FORWARD -p tcp -d 10.0.2.2 --dport 25 -j ACCEPT
```

```
iptables -A FORWARD -p tcp -s 10.0.2.2 --sport 25 -j ACCEPT
```

e. 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.

```
iptables -I INPUT -s 192.56.0.11 -j DROP
```

```
iptables -I INPUT -s 14.12.44.0/18 -j DROP
```

```
iptables -I FORWARD -s 192.56.0.11 -j DROP
```

```
iptables -I FORWARD -s 14.12.44.0/18 -j DROP
```

f. Запрещают доступ к ssh серверу на c7-1 из внешней сети.

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

g. Разрешает доступ к ssh серверу на c7-1 из внутренней сети.

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

h. Разрешает icmp эхо запросы из внутренней сети наружу только на хост 8.8.8.8

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

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

i. Запрещает хосту c7-1 давать icmp эхо ответы, но при этом сохраняет возможность с самого хоста c7-1 делать icmp эхо запросы и получать на них ответы.

```
iptables -A INPUT -p icmp --icmp-type echo-request -j ACCEPT
```

```
iptables -A INPUT -p icmp --icmp-type echo-reply -j ACCEPT
```

```
iptables -A OUTPUT -p icmp --icmp-type echo-reply -j DROP
```

7. Доступ через ssh защищенным сервисам

```
ssh -p 1334 -g -L 8888:localhost:80 KEGuser@localhost
```

Ответы на вопросы:

1. **Отличие MASQUERADE от SNAT:**

MASQUERADE. Менее быстрая маршрутизация чем SNAT при массовых запросах, так как для каждого нового соединения определяется IP адрес на внешнем сетевом интерфейсе (WAN). Отлично подходит для маршрутизаторов домашнего использования и когда меняется IP адрес на WAN интерфейсе.
SNAT. Если IP адрес на внешнем сетевом интерфейсе (WAN) статический и не меняется, то желательно использовать именно SNAT. В правилах фаервола указывается один или несколько IP адресов, таким образом есть возможность распределить локальные IP адреса и сети по внешним IP адресам. SNAT хорошо использовать на серверах доступа.

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

Таблица filter

Основная таблица, служит для фильтрации пакетов, именно здесь происходит принятие решений о разрешении или запрете дальнейшего движения пакета в системе. Используется по умолчанию, если явно не указано имя другой таблицы. Содержит цепочки INPUT, FORWARD и OUTPUT.

Таблица nat

Используется для трансляции сетевых адресов, т.е. подмены адреса получателя/отправителя, применяется, если сервер используется в качестве маршрутизатора. Содержит цепочки PREROUTING, OUTPUT, POSTROUTING.

Таблица raw

Содержит цепочки PREROUTING и OUTPUT, здесь производятся манипуляции с пакетами до задействования механизма определения состояний.

Таблица mangle

Предназначена для модификации заголовков сетевых пакетов, таких параметров как ToS (Type of Service), TTL (Time To Live), MARK. Содержит все существующие пять цепочек.

Таблица security

Используется для взаимодействия с внешними системами безопасности, в частности с SELinux и AppArmor. Содержит цепочки INPUT, OUTPUT и FORWARD.

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

Iptables позволяет создавать пользовательские цепочки для более гибкого управления правилами. Например, можно создать цепочку для обработки всего трафика от определенного IP-адреса:

```
iptables -N MY_CHAIN iptables -A MY_CHAIN -s 192.168.0.100 -j DROP iptables -A INPUT -j MY_CHAIN
```

В этом примере команда -N MY_CHAIN создает новую цепочку с именем MY_CHAIN. Затем добавляется правило, которое отбрасывает все пакеты от 192.168.0.100. Наконец, цепочка MY_CHAIN добавляется в цепочку INPUT, что означает, что все входящие пакеты будут также проходить через цепочку MY_CHAIN.

4. **Имеет ли смысл порядок правил?**

Да, имеет. Попав в цепочку пакет последовательно проходит правила в порядке их перечисления до первого срабатывания. Дальнейшее его движение зависит от типа действия, если действие нетерминальное - пакет продолжит движение по цепочке, иначе – покидает ее

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**

```
iptables -A INPUT -m state --state ESTABLISHED, RELATED -j ACCEPT
```

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

Вывод: в результате лабораторной работы закреплено понимание принципов работы NAT и firewall, а также сформировали начальные навыки в конфигурировании NAT и Firewall на платформе Linux