Телекоммуникационные системы и технологии

Лабораторная работа №6

Трансляция адресов в ОС Linux

Выполнил: Птицын Владислав
Группа: М3301
Преподаватель: Береснев А.Д.

Санкт-Петербург, 2024

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

Измененные параметры sshd из Части 2

```
PermitRootLogin no
MaxAuthTries 2
LoginGraceTime 30s
UseDNS no
```

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

```
# Generated by iptables-save v1.4.21 on Mon Nov  4 18:42:38 2024
*nat
:PREROUTING ACCEPT [0:0]
:INPUT ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
:POSTROUTING ACCEPT [0:0]
-A PREROUTING -i enp0s3 -p tcp -m tcp --dport 55022 -j DNAT --to-destination 10.0.0.2:22
-A POSTROUTING -s 10.0.0.0/24 -o enp0s3 -j SNAT --to-source 10.0.2.15
COMMIT
# Completed on Mon Nov  4 18:42:38 2024
# Generated by iptables-save v1.4.21 on Mon Nov  4 18:42:38 2024
*filter
:INPUT ACCEPT [281:24343]
:FORWARD DROP [0:0]
:OUTPUT ACCEPT [187:21655]
-A FORWARD -i enp0s8 -o enp0s3 -j ACCEPT
-A FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT
-A FORWARD -i enp0s3 -o enp0s8 -j ACCEPT
COMMIT
# Completed on Mon Nov  4 18:42:38 2024
```

```
# sample configuration for iptables service
# you can edit this manually or use system-config-firewall
# please do not ask us to add additional ports/services to this default configuration
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -p icmp -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 22 -j ACCEPT
-A INPUT -j REJECT --reject-with icmp-host-prohibited
-A FORWARD -j REJECT --reject-with icmp-host-prohibited
COMMIT
```

Команду и консольный вывод из Части 4 п.3

```
nmap 10.0.0.2 | grep open
```

```
22/tcp open  ssh
80/tcp open  http
```

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

Пункт 1

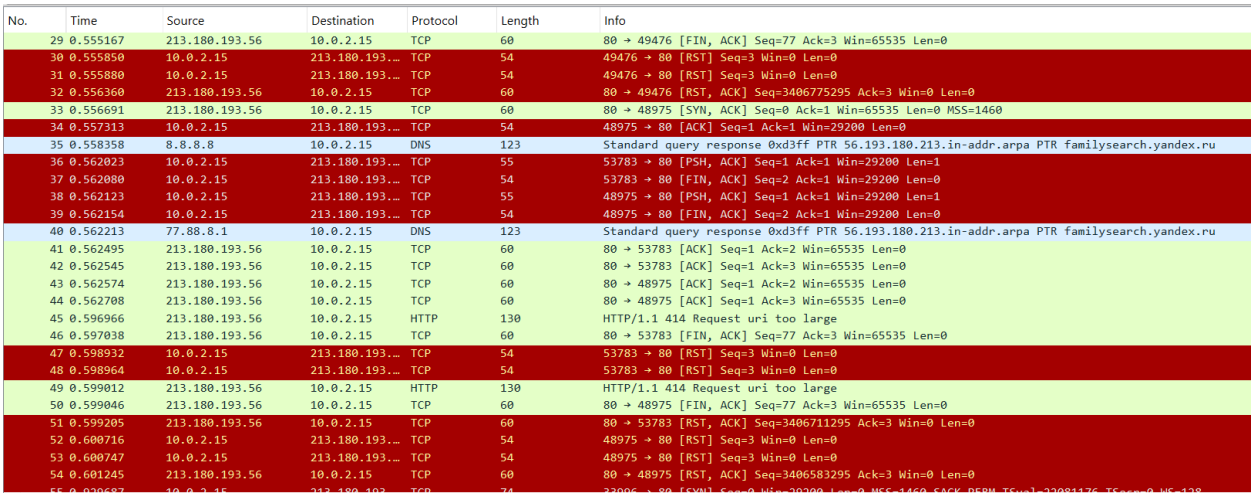
```
ss -tan
```

```
State      Recv-Q Send-Q Local Address:Port               Peer Address:Port
LISTEN     0      128          *:80                       *:*
LISTEN     0      128          *:22                       *:*
LISTEN     0      100    127.0.0.1:25                       *:*
TIME-WAIT  0      0      10.0.0.2:80                10.0.0.1:42808
ESTAB      0      0      10.0.0.2:22                10.0.2.2:62035
LISTEN     0      128       [::]:22                    [::]:*
LISTEN     0      100      [::1]:25                    [::]:*
```

Команды пунктов 4,6,8

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

tcpdump -i enp0s3 -nn -w ext.pcap
tcpdump -i enp0s8 -nn -w int.pcap

tcpdump -i enp0s3 -nn -w temp.pcap
```

Пункт 4
Трафик на внешнем интерфейсе



No.	Time	Source	Destination	Protocol	Length	Info
29	0.555167	213.180.193.56	10.0.2.15	TCP	60	80 → 49476 [FIN, ACK] Seq=77 Ack=3 Win=65535 Len=0
30	0.555850	10.0.2.15	213.180.193.…	TCP	54	49476 → 80 [RST] Seq=3 Win=0 Len=0
31	0.555880	10.0.2.15	213.180.193.…	TCP	54	49476 → 80 [RST] Seq=3 Win=0 Len=0
32	0.556360	213.180.193.56	10.0.2.15	TCP	60	80 → 49476 [RST, ACK] Seq=3406775295 Ack=3 Win=0 Len=0
33	0.556691	213.180.193.56	10.0.2.15	TCP	60	80 → 48975 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460
34	0.557313	10.0.2.15	213.180.193.…	TCP	54	48975 → 80 [ACK] Seq=1 Ack=1 Win=29200 Len=0
35	0.558358	8.8.8.8	10.0.2.15	DNS	123	Standard query response 0xd3ff PTR 56.193.180.213.in-addr.arpa PTR familysearch.yandex.ru
36	0.562023	10.0.2.15	213.180.193.…	TCP	55	53783 → 80 [PSH, ACK] Seq=1 Ack=1 Win=29200 Len=1
37	0.562080	10.0.2.15	213.180.193.…	TCP	54	53783 → 80 [FIN, ACK] Seq=2 Ack=1 Win=29200 Len=0
38	0.562123	10.0.2.15	213.180.193.…	TCP	55	48975 → 80 [PSH, ACK] Seq=1 Ack=1 Win=29200 Len=1
39	0.562154	10.0.2.15	213.180.193.…	TCP	54	48975 → 80 [FIN, ACK] Seq=2 Ack=1 Win=29200 Len=0
40	0.562213	77.88.8.1	10.0.2.15	DNS	123	Standard query response 0xd3ff PTR 56.193.180.213.in-addr.arpa PTR familysearch.yandex.ru
41	0.562495	213.180.193.56	10.0.2.15	TCP	60	80 → 53783 [ACK] Seq=1 Ack=2 Win=65535 Len=0
42	0.562545	213.180.193.56	10.0.2.15	TCP	60	80 → 53783 [ACK] Seq=1 Ack=3 Win=65535 Len=0
43	0.562574	213.180.193.56	10.0.2.15	TCP	60	80 → 48975 [ACK] Seq=1 Ack=2 Win=65535 Len=0
44	0.562708	213.180.193.56	10.0.2.15	TCP	60	80 → 48975 [ACK] Seq=1 Ack=3 Win=65535 Len=0
45	0.596966	213.180.193.56	10.0.2.15	HTTP	130	HTTP/1.1 414 Request uri too large
46	0.597038	213.180.193.56	10.0.2.15	TCP	60	80 → 53783 [FIN, ACK] Seq=77 Ack=3 Win=65535 Len=0
47	0.598932	10.0.2.15	213.180.193.…	TCP	54	53783 → 80 [RST] Seq=3 Win=0 Len=0
48	0.598964	10.0.2.15	213.180.193.…	TCP	54	53783 → 80 [RST] Seq=3 Win=0 Len=0
49	0.599012	213.180.193.56	10.0.2.15	HTTP	130	HTTP/1.1 414 Request uri too large
50	0.599046	213.180.193.56	10.0.2.15	TCP	60	80 → 48975 [FIN, ACK] Seq=77 Ack=3 Win=65535 Len=0
51	0.599205	213.180.193.56	10.0.2.15	TCP	60	80 → 53783 [RST, ACK] Seq=3406711295 Ack=3 Win=0 Len=0
52	0.600716	10.0.2.15	213.180.193.…	TCP	54	48975 → 80 [RST] Seq=3 Win=0 Len=0
53	0.600747	10.0.2.15	213.180.193.…	TCP	54	48975 → 80 [RST] Seq=3 Win=0 Len=0
54	0.601245	213.180.193.56	10.0.2.15	TCP	60	80 → 48975 [RST, ACK] Seq=3406583295 Ack=3 Win=0 Len=0

Видим как подменяется адрес 10.0.0.2 на 10.0.0.15

Трафик на внутреннем интерфейсе

No.	Time	Source	Destination	Protocol	Length	Info
13	0.221969	10.0.0.2	8.8.8.8	DNS	81	Standard query 0x89c7 PTR 2.2.0.10.in-addr.arpa
14	0.222052	10.0.0.2	77.88.8.1	DNS	81	Standard query 0x89c7 PTR 2.2.0.10.in-addr.arpa
15	0.226365	8.8.8.8	10.0.0.2	DNS	81	Standard query response 0x89c7 No such name PTR 2.2.0.10.in-addr.arpa
16	0.227792	77.88.8.1	10.0.0.2	DNS	81	Standard query response 0x89c7 No such name PTR 2.2.0.10.in-addr.arpa
17	0.319575	10.0.0.2	213.180.193.…	TCP	74	49476 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM TSval=22080566 TSecr=0 WS=128
18	0.348943	213.180.193.56	10.0.0.2	TCP	58	80 → 49476 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460
19	0.349931	10.0.0.2	213.180.193.…	TCP	60	49476 → 80 [ACK] Seq=1 Ack=1 Win=29200 Len=0
20	0.421725	10.0.0.2	213.180.193.…	TCP	74	53783 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM TSval=22080668 TSecr=0 WS=128
21	0.455206	213.180.193.56	10.0.0.2	TCP	58	80 → 53783 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460
22	0.456174	10.0.0.2	213.180.193.…	TCP	60	53783 → 80 [ACK] Seq=1 Ack=1 Win=29200 Len=0
23	0.522463	10.0.0.2	213.180.193.…	TCP	60	49476 → 80 [PSH, ACK] Seq=1 Ack=1 Win=29200 Len=1
24	0.522565	10.0.0.2	213.180.193.…	TCP	60	49476 → 80 [FIN, ACK] Seq=2 Ack=1 Win=29200 Len=0
25	0.523525	213.180.193.56	10.0.0.2	TCP	54	80 → 49476 [ACK] Seq=1 Ack=2 Win=65535 Len=0
26	0.523593	213.180.193.56	10.0.0.2	TCP	54	80 → 49476 [ACK] Seq=1 Ack=3 Win=65535 Len=0
27	0.524084	10.0.0.2	8.8.8.8	DNS	87	Standard query 0xd3ff PTR 56.193.180.213.in-addr.arpa
28	0.524172	10.0.0.2	77.88.8.1	DNS	87	Standard query 0xd3ff PTR 56.193.180.213.in-addr.arpa
29	0.525058	10.0.0.2	213.180.193.…	TCP	74	48975 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM TSval=22080772 TSecr=0 WS=128
30	0.555184	213.180.193.56	10.0.0.2	HTTP	130	HTTP/1.1 414 Request uri too large
31	0.555227	213.180.193.56	10.0.0.2	TCP	54	80 → 49476 [FIN, ACK] Seq=77 Ack=3 Win=65535 Len=0
32	0.555871	10.0.0.2	213.180.193.…	TCP	60	49476 → 80 [RST] Seq=3 Win=0 Len=0
33	0.555919	10.0.0.2	213.180.193.…	TCP	60	49476 → 80 [RST] Seq=3 Win=0 Len=0
34	0.556768	213.180.193.56	10.0.0.2	TCP	58	80 → 48975 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460
35	0.557330	10.0.0.2	213.180.193.…	TCP	60	48975 → 80 [ACK] Seq=1 Ack=1 Win=29200 Len=0
36	0.558484	8.8.8.8	10.0.0.2	DNS	123	Standard query response 0xd3ff PTR 56.193.180.213.in-addr.arpa PTR familysearch.yandex.ru
37	0.562026	10.0.0.2	213.180.193.…	TCP	60	53783 → 80 [PSH, ACK] Seq=1 Ack=1 Win=29200 Len=1
38	0.562110	10.0.0.2	213.180.193.…	TCP	60	53783 → 80 [FIN, ACK] Seq=2 Ack=1 Win=29200 Len=0

Видим, что адреса 10.0.0.1 нет, так как пакеты сразу же транслируются на внешний интерфейс

Пункт 8

```
20:30:53.576573 IP 10.0.2.2.62391 > 10.0.0.2.22: Flags [S], seq 929856001, win 65535, options [mss 1460], length 0
20:30:53.576613 IP 10.0.0.2.22 > 10.0.2.2.62391: Flags [S.], seq 4144259214, ack 929856002, win 29200, options [mss 1460], length 0
20:30:53.577244 IP 10.0.2.2.62391 > 10.0.0.2.22: Flags [.], ack 1, win 65535, length 0
20:30:53.582226 IP 10.0.2.2.62391 > 10.0.0.2.22: Flags [P.], seq 1:34, ack 1, win 65535, length 33
20:30:53.582247 IP 10.0.0.2.22 > 10.0.2.2.62391: Flags [.], ack 34, win 29200, length 0
```

Видим как сначала синхронизация S (SYN), потом отправка данных P (PSH)

```
20:31:09.455964 IP 10.0.2.2.62391 > 10.0.0.2.22: Flags [.], ack 3490, win 65535, length 0
20:31:09.457890 IP 10.0.2.2.62391 > 10.0.0.2.22: Flags [P.], seq 2814:2910, ack 3490, win 65535, length 96
20:31:09.458323 IP 10.0.2.2.62391 > 10.0.0.2.22: Flags [R.], seq 2910, ack 3490, win 65535, length 0
```

В конце передачи видим как был послан флаг R (RST)
В Wireshark видим ещё и обмен ключами

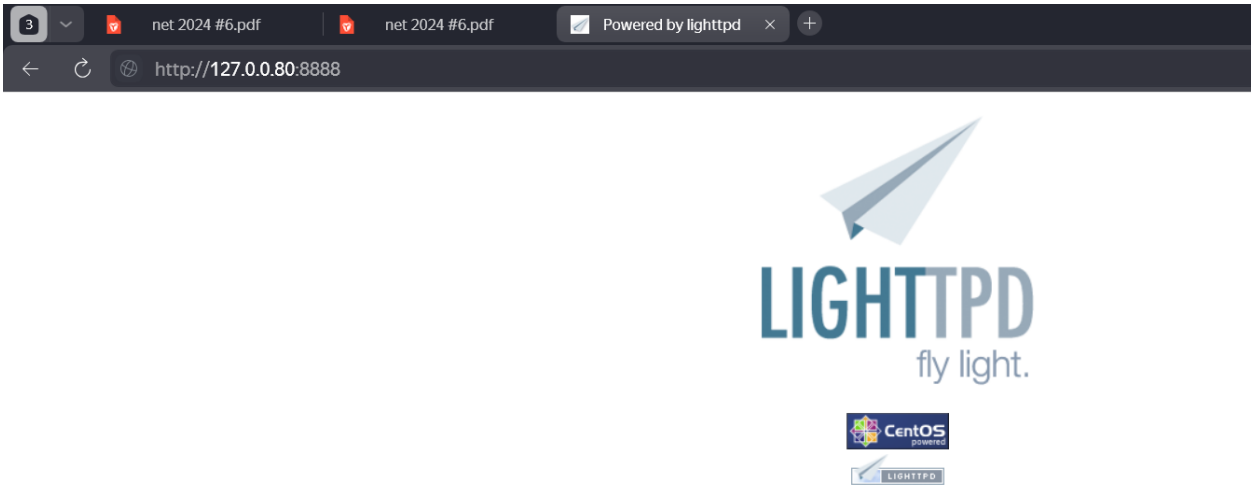
```
22 → 62392 [ACK] Seq=1 Ack=34 Win=29200 Len=0
Server: Protocol (SSH-2.0-OpenSSH_7.4)
62392 → 22 [ACK] Seq=34 Ack=22 Win=65535 Len=0
Server: Key Exchange Init
Client: Key Exchange Init
62392 → 22 [ACK] Seq=1466 Ack=1302 Win=65535 Len=0
Client: Elliptic Curve Diffie-Hellman Key Exchange Init
22 → 62392 [ACK] Seq=1302 Ack=1514 Win=31504 Len=0
Server: Elliptic Curve Diffie-Hellman Key Exchange Reply, New Keys, Encrypted packet (len=84)
62392 → 22 [ACK] Seq=1514 Ack=1594 Win=65535 Len=0
Client: New Keys
Client: Encrypted packet (len=44)
22 → 62392 [ACK] Seq=1594 Ack=1574 Win=31504 Len=0
Server: Encrypted packet (len=44)
62392 → 22 [ACK] Seq=1574 Ack=1638 Win=65535 Len=0
Client: Encrypted packet (len=68)
Server: Encrypted packet (len=84)
```

Текст итоговых правил iptables с с7-1.

```
[H[JChain PREROUTING (policy ACCEPT 2178 packets, 139K bytes)
 pkts bytes target     prot opt in     out     source               destination
   54  2376 DNAT       tcp  --  enp0s3 *       0.0.0.0/0            0.0.0.0/0           tcp dpt:55022 to:10.0.0.2:22

Chain INPUT (policy ACCEPT 34 packets, 2048 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 48 packets, 3340 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain POSTROUTING (policy ACCEPT 67 packets, 4176 bytes)
 pkts bytes target     prot opt in     out     source               destination
  266 16508 SNAT       all  --  *      enp0s3  10.0.0.0/24          0.0.0.0/0           to:10.0.2.15
Chain INPUT (policy DROP 15 packets, 788 bytes)
 pkts bytes target     prot opt in     out     source               destination
    2   168 ACCEPT     icmp --  *      *       0.0.0.0/0            0.0.0.0/0
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            10.0.0.0/24         tcp dpt:22
   27  1400 ACCEPT     tcp  --  *      *       10.0.0.0/24          0.0.0.0/0           tcp dpt:22
    0     0 DROP       all  --  *      *       14.12.0.0/18         0.0.0.0/0
    0     0 DROP       all  --  *      *       192.56.0.11          0.0.0.0/0
    0     0 ACCEPT     tcp  --  *      *       192.168.1.0/24       0.0.0.0/0           tcp dpt:25

Chain FORWARD (policy DROP 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 DROP       icmp --  *      *       0.0.0.0/0            0.0.0.0/0
    2   168 ACCEPT     icmp --  *      *       8.8.8.8              10.0.0.0/24
    5   420 ACCEPT     icmp --  *      *       10.0.0.0/24          8.8.8.8
    0     0 DROP       all  --  *      *       14.12.0.0/18         0.0.0.0/0
    0     0 DROP       all  --  *      *       192.56.0.11          0.0.0.0/0
  240 36375 ACCEPT     tcp  --  *      *       0.0.0.0/0            10.0.0.2            tcp dpt:22 state NEW,ESTABLISHED
  150 33135 ACCEPT     tcp  --  *      *       10.0.0.2             0.0.0.0/0           tcp spt:22 state NEW,ESTABLISHED
    0     0 ACCEPT     udp  --  *      *       8.8.8.8              10.0.2.15           udp spt:53
    0     0 ACCEPT     udp  --  *      *       77.88.8.1            10.0.2.15           udp spt:53
    0     0 ACCEPT     tcp  --  *      *       8.8.8.8              10.0.2.15           tcp spt:53
    0     0 ACCEPT     tcp  --  *      *       77.88.8.1            10.0.2.15           tcp spt:53
    0     0 ACCEPT     udp  --  *      *       10.0.0.0/24          8.8.8.8             udp dpt:53
    0     0 ACCEPT     udp  --  *      *       10.0.0.0/24          77.88.8.1           udp dpt:53
    0     0 ACCEPT     tcp  --  *      *       10.0.0.0/24          8.8.8.8             tcp dpt:53
    0     0 ACCEPT     tcp  --  *      *       10.0.0.0/24          77.88.8.1           tcp dpt:53
    0     0 ACCEPT     tcp  --  *      *       10.0.0.0/24          0.0.0.0/0           tcp dpt:110
    0     0 ACCEPT     tcp  --  *      *       10.0.0.0/24          0.0.0.0/0           tcp dpt:80
    0     0 ACCEPT     tcp  --  *      *       10.0.0.0/24          0.0.0.0/0           tcp dpt:443
    0     0 ACCEPT     tcp  --  *      *       10.0.0.0/24          0.0.0.0/0           tcp dpt:8080
    0     0 ACCEPT     tcp  --  *      *       10.0.0.0/24          0.0.0.0/0           tcp dpt:22
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            10.0.2.15           tcp spt:22
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            10.0.2.15           tcp spt:443
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            10.0.2.15           tcp spt:80
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            10.0.2.15           tcp spt:8080
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            10.0.2.15           tcp spt:110

Chain OUTPUT (policy ACCEPT 5 packets, 348 bytes)
 pkts bytes target     prot opt in     out     source               destination
```

Команду подключения из Части 7, п.1.

```
ssh -N -o GatewayPorts=yes -L 127.0.0.80:8888:10.0.0.2:80 PVAuser@127.0.0.4 -p 2222
```



Вопросы и задания:

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

SNAT – руками ставим статический ip, который будет source в исходящем из NAT пакете
MASQUERADE – динамический ip

Используйте SNAT, когда у вас есть статический внешний IP-адрес, который не меняется.
Используйте MASQUERADE, когда у вас есть динамический внешний IP-адрес, который может изменяться.

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

Таблица filter:
INPUT, FORWARD, OUTPUT

Таблица nat:
PREROUTING, INPUT, OUTPUT, POSTROUTING

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

```
iptables -N TEMP
iptables -A INPUT -j TEMP
```

4) Имеет ли смысл порядок правил?

Да имеет, после первого подходящего правила будет выполнено действие в колонке TARGET и пакет будет изъят из цепочки

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