



NAT - реализация с помощью PF

Опубликовано muff в Пт, 2010-06-18 13:40

Решил "пощупать" реализацию NAT с помощью PF. Ознакомившись с документацией, убедился, что все не так и сложно.

Итак, исходные данные:

- офисная сеть 10.200.106.0/24;
- интерфейс vlan106 - "смотрит" в офисную сеть;
- интерфейс vlan684 - "смотрит" в Internet, default;
- интерфейс vlan685 - "смотрит" в UA-IX, получает от вышестоящего маршрутизатора список сетей UA-IX.

Исходя из указанного построения сети, понятно, что NATить необходимо на двух внешних интерфейсах одновременно (на vlan684 и vlan685 соответственно).

Итак, для начала отредактируем конфигурационный файл **pf.conf** под свои нужды. В моем случае конфигурация получилась следующая:

```
# cat /etc/pf.conf

# Задаем количество записей в таблице состояний
set limit states 500000
# Указываем значение таймаутов
set optimization aggressive
# NAT на интерфейсе vlan684
nat pass on vlan684 from 10.200.106.0/24 to any -> vlan684
# NAT на интерфейсе vlan685
nat pass on vlan685 from 10.200.106.0/24 to any -> vlan685
```

????????? ? ?????????? ??????? rc.conf ?????? PF:

```
# echo '# Packet Filter Firewall' >> /etc/rc.conf
# echo 'pf_enable="YES"' >> /etc/rc.conf
```

Запускаем Packet Filter:

```
# sh /etc/rc.d/pf start

Enabling pf.
No ALTQ support in kernel
ALTQ related functions disabled
No ALTQ support in kernel
ALTQ related functions disabled
No ALTQ support in kernel
ALTQ related functions disabled
pf enabled
```



На всякий случай "принудительно" перечитываем конфигурационный файл:

```
# pfctl -f /etc/pf.conf
```

```
No ALTQ support in kernel
ALTQ related functions disabled
```

Проверяем tcpdump-ом, работает ли NAT:

```
# tcpdump -i vlan106
```

```
tcpdump: verbose output suppressed, use -v or -vv for full protocol decoding
listening on vlan106, link-type EN10MB (Ethernet), capture size 68 bytes
14:56:19.105705 IP 10.200.106.3.1914 > ip254-116.starnet.lv.7777: . ack 2441059928 win 6528514:56:19.27962
2 IP ip254-116.starnet.lv.7777 > 10.200.106.3.1914: P 1:12(11) ack 0 win 6502214:56:19.406627
IP 10.200.106.3.1914 > ip254-116.starnet.lv.7777: . ack 12 win 6527414:56:19.535325
IP 10.200.106.19.2282 > 194-182-17-190.fibertel.com.ar.6881: S 3138508329:3138508329(0) win 65535 <mss 1460,nop,nop,sackOK>
14:56:19.901696 IP ip254-116.starnet.lv.7777 > 10.200.106.3.1914: P 12:23(11) ack 0 win 6502214:56:20.008502
IP 10.200.106.3.1914 > ip254-116.starnet.lv.7777: . ack 23 win 6526314:56:20.130111
IP 10.200.106.19.6881 > 110-196-16-190.fibertel.com.ar.6881: UDP, length 6514:56:20.181766
IP ip254-116.starnet.lv.7777 > 10.200.106.3.1914: P 23:34(11) ack 0 win 65022
```

Трафик на внутреннем интерфейсе идет в обе стороны. Соответственно все в норме. Для проверки состояния воспользуемся следующей командой:

```
# pfctl -si
```

```
No ALTQ support in kernel
ALTQ related functions disabled
Status: Enabled for 0 days 00:35:52
Debug: Urgent
```

State Table	Total	Rate	current entries
2085 searches	88476828	41113.8/s	insert
106714	108799	50.6/s	removals
37935.3/s bad-offset	Counters match	81636835	
16	0	0.0/s	fragment
0.0/s normalize	short	0	0.0/s
0	0	0.0/s	memory
0	0.0/s	bad-timestamp	0.0/s
0	0.0/s	congestion	0
0	8	0.0/s	proto-cksum
219	0.1/s	state-mismatch	152
0.1/s state-insert	0	0.0/s	state-limit
0	0.0/s	src-limit	0
0	0.0/s	synproxy	0
			0.0/s

Как оказалось, ничего сложного...

Источник (получено 2026-04-11 22:18):

<http://muff.kiev.ua/content/nat-realizatsiya-s-pomoshchyu-pf>