Настройка сети и основы маршрутизации в Linux

УЦ "Специалист"

Приложение к курсу «Основы администрирования и безопасности Linux»

Утилиты для работы с сетью

- *ifconfig* – просмотр и настройка: сетевых интерфейсов, ip-адресов и mac-адресов
- *route* – просмотр и настройка таблицы маршрутизации
- *arp* – просмотр и настройка таблицы соответствия mac и ip-адресов
- *ping* – утилита для проверки доступности хостов в сети
- *traceroute* – утилита для ослеживания маршрута от одного хоста до другого
- *netstat* – просмотр статиститки по сетевым интерфейсам, отчетов по сетевым подключениям, службам и маршрутизации пакетов
- *nslookup* – позволяет взаимодействовать с DNS-серверами
- *nmap* – сканер портов на предмет поиска уязвимостей, с целью их устранения.

ifconfig

- *ifconfig* – просмотр и настройка: сетевых интерфейсов, ip-адресов и mac-адресов
- Опции:
  - имя устройства (сетевого интерфейса - eth0,eth1...)
  - IP-адрес интерфейса
  - маска подсети
  - широковещательный адрес
- Пример:
  - *ifconfig eth0 10.10.103.230 netmask 255.255.255.0 broadcast 10.10.103.255*

route

- *route* – просмотр и настройка таблицы маршрутизации
- Опции:
  - *add* – добавление маршрута в таблицу маршрутизации
    - *-net* – добавдение маршрута к сети
    - *-host* – добавление маршрута к хосту
    - *default* – добавление маршрута по умолчанию
  - *del* – удаление маршрута из таблицы
  - *gw* – указание адреса шлюза
- Пример:
  - *route add default gw 10.10.103.100*

arp

- *arp* – просмотр и настройка таблицы соответствия mac и ip-адресов
- Опции:
  - *a [hostname]* – показывает значение соответствия mac и ip-адреса для указанного хоста. Если не указать хост, будут показаны все значения таблицы.
  - *d hostname* – удаляет запись из таблицы
  - *s hostname mac* – вручную добавляет запись в таблицу
- Пример:
  - *termsrv:~ # arp -a*

    ivanova (192.168.213.24) at 00:1A:4D:41:0F:F5 [ether] on eth0

    petroff (192.168.213.213) at 00:18:71:71:96:66 [ether] on eth0

    sidorov (192.168.213.89) at 00:1A:4D:41:09:DF [ether] on eth0

ping

- *ping* – утилита для проверки доступности хостов в сети
- Опции:

  -R – включить опцию сохранения маршрута в передваемых пакетах
  -b – разрешить широковещательную рассылку
  -c – ограничить число отправляемых пакетов
  -i – установить интервал между отправкой пакетов (по умолчанию 1 секунда)
  -s – установить размер пакета (по умолчанию 56 байт)

- Пример:

```
ping -c4 www.1web.ru
PING www.1web.ru (213.152.131.199) 56(84) bytes of data.
64 bytes from s2.stability.ru (213.152.131.199): icmp_seq=1 ttl=58 time=5.55 ms
64 bytes from s2.stability.ru (213.152.131.199): icmp_seq=2 ttl=58 time=9.66 ms
64 bytes from s2.stability.ru (213.152.131.199): icmp_seq=3 ttl=58 time=4.93 ms
64 bytes from s2.stability.ru (213.152.131.199): icmp_seq=4 ttl=58 time=4.83 ms

--- www.1web.ru ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 4.833/6.248/9.669/1.994 ms
```

Установка traceroute

Данная утилита не устанавливается по умолчанию, поэтому надо ее доустановить:

*sudo apt-get install tracerout**

traceroute

- *traceroute* – утилита для ослеживания маршрута от одного хоста до другого
- Опции:
  -n – отключить преобразование ip-адресов в DNS-имена
  -m – установка максимального количества контрольных точек (хопов) через которые пройдет отправленный пакет (по умолчанию 30)
- Пример:

traceroute -n www.1web.ru
traceroute to www.1web.ru (213.152.131.199), 30 hops max, 40 byte packets
1 192.168.1.1 (192.168.1.1) 1.133 ms 1.415 ms 1.882 ms
2 213.219.200.4 (213.219.200.4) 5.898 ms 6.945 ms 8.785 ms
3 213.219.200.1 (213.219.200.1) 9.552 ms 11.449 ms 8.424 ms
4 193.232.244.209 (193.232.244.209) 10.238 ms 13.247 ms 11.186 ms
5 213.152.128.81 (213.152.128.81) 12.578 ms 15.621 ms 16.070 ms
6 213.152.131.199 (213.152.131.199) 17.009 ms 16.889 ms 18.970 ms

netstat

- *netstat* – просмотр статиститки по сетевым интерфейсам, отчетов по сетевым подключениям, службам и маршрутизации пакетов
- Опции:
  - -n - отключить преобразование ip-адресов в DNS-имена
  - -l – показать порты, открытые для прослушивания
  - -i – показать статистику по сетевым интерфейсам
  - -r – показать таблицу маршрутизации
  - -s – показать статистику по каждому протоколу
  - -p – показывает имя и PID-программы
- Пример:

```
netstat -i
Kernel Interface table
Iface   MTU Met   RX-OK RX-ERR RX-DRP RX-OVR   TX-OK TX-ERR TX-DRP TX-OVR Flg
eth0   1500   0       0      0      0      0       0      0      0      0 BMU
eth1   1500   0  639668      0      0      0  445915      0      0      0 BMNRU
lo    16436   0     617      0      0      0     617      0      0      0 LRU
```

nslookup

- *nslookup* – позволяет взаимодействовать с DNS-серверами
- Пример:

nslookup www.specialist.ru

Server: 10.0.0.1
Address: 10.0.0.1#53

Non-authoritative answer:
www.specialist.ru canonical name = websrv.specialist.ru.
Name: websrv.specialist.ru
Address: 213.189.207.228

Установка nmap

Данная утилита не устанавливается по умолчанию, поэтому надо ее доустановить:

sudo apt-get install nmap

nmap

nmap – сканер портов на предмет поиска уязвимостей, с целью их устранения.

- Опции:
  - A – включить распознавание ОС и ее версии
  - sU – сканировать UDP-порты
  - SO – сканировать TCP-порты
- Пример:

nmap -A my.router

Starting Nmap 4.20 (http://insecure.org) at 2008-08-26 14:39 MSD

Interesting ports on my.router (10.0.11.18):
Not shown: 1692 closed ports
PORT STATE SERVICE VERSION
22/tcp open sshd
53/tcp open domain ISC Bind dnsmasq-2.22
80/tcp open http Linksys wireless-G WAP http config (Name WL500g.Deluxe)
5000/tcp open UPnP?
9100/tcp open jetdirect?

Service detection performed. Please report any incorrect results at http://insecure.org/nmap/submit/ .
Nmap finished: 1 IP address (1 host up) scanned in 111.471 seconds

Утилиты для работы с сетью

- *ifconfig* – просмотр и настройка: сетевых интерфейсов, ip-адресов и mac-адресов
- *route* – просмотр и настройка таблицы маршрутизации
- *arp* – просмотр и настройка таблицы соответствия mac и ip-адресов
- *ping* – утилита для проверки доступности хостов в сети
- *traceroute* – утилита для ослеживания маршрута от одного хоста до другого
- *netstat* – просмотр статиститки по сетевым интерфейсам, отчетов по сетевым подключениям, службам и маршрутизации пакетов
- *nslookup* – позволяет взаимодействовать с DNS-серверами
- *nmap* – сканер портов на предмет поиска уязвимостей, с целью их устранения.

Настройка сети

```
# sudo nano /etc/init.d/rc.local
ifconfig eth0 10.10.103.X netmask 255.255.255.0 \
broadcast 10.10.103.255 up
route add default gw 10.10.103.100
echo “search unix.specialist.ru”  >  /etc/resolv.conf
echo “nameserver 10.10.103.100”  >>  /etc/resolv.conf
hostname c230
```

Внимание! Символ «\» означает, что команда должна писаться в одну строку.

Проверка работы сети

Запустите скрипт /etc/init.d/rc.local и выполните следующие проверки:

ping внешний_ip_соседа

ping внутренний_ip_соседа

nslookup ya.ru

Если ни одна из команд не сработала, возможно у вас внешний интерфейс eth1, а внутрений — eth0, тогда настройки будут следующие:

sudo bash

ifconfig eth0 10.10.103.X netmask 255.255.255.0 broadcast 10.10.103.255 up

ifconfig eth1 192.168.Y.Z netmask 255.255.255.0 broadcast 192.168.2.255 up

route add default gw 10.10.103.100

echo “search unix.specialist.ru” > /etc/resolv.conf

echo “nameserver 10.10.103.100” >> /etc/resolv.conf

hostname c230

Присвоение нескольких IP на один сетевой интерфейс

Для присвоения еще одного IP адреса интерфейсу eth0, необходимо использовать программу ifconfig и интерфесы: eth0:0, eth0:1, etth0:2 и т.д.

sudo ifconfig ethX:Y IP netmask MASK broadcast BROADCAST_IP up

где

IP — IP-адрес, например 172.16.0.1

MASK — маска подсети, например 255.255.255.0

BROADCAST_IP — адрес широковещательной рассылки, например 172.16.0255

Пример использования второго IP-адреса:

sudo ifconfig eth0:0 172.16.0.X netmask 255.255.255.0 broadcast 172.16.0.255 up

Проверьте работу второго интерфейса следующей командой:

ping 172.16.0.N

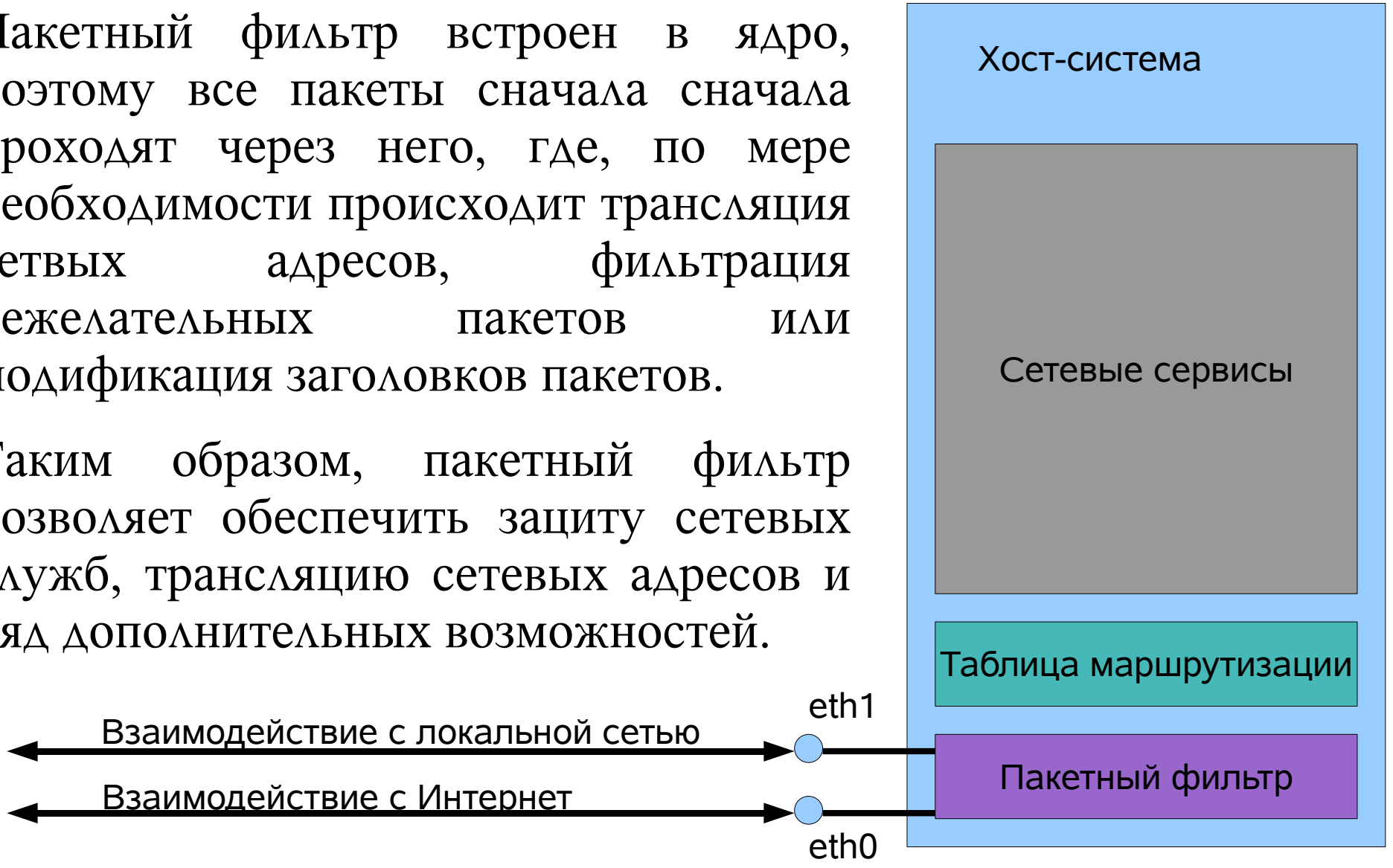
где

N — последняя цифра IP-адреса соседа

Пакетный фильтр

Пакетный фильтр встроен в ядро, поэтому все пакеты сначала сначала проходят через него, где, по мере необходимости происходит трансляция сетвых адресов, фильтрация нежелательных пакетов или модификация заголовков пакетов.

Таким образом, пакетный фильтр позволяет обеспечить защиту сетевых служб, трансляцию сетевых адресов и ряд дополнительных возможностей.



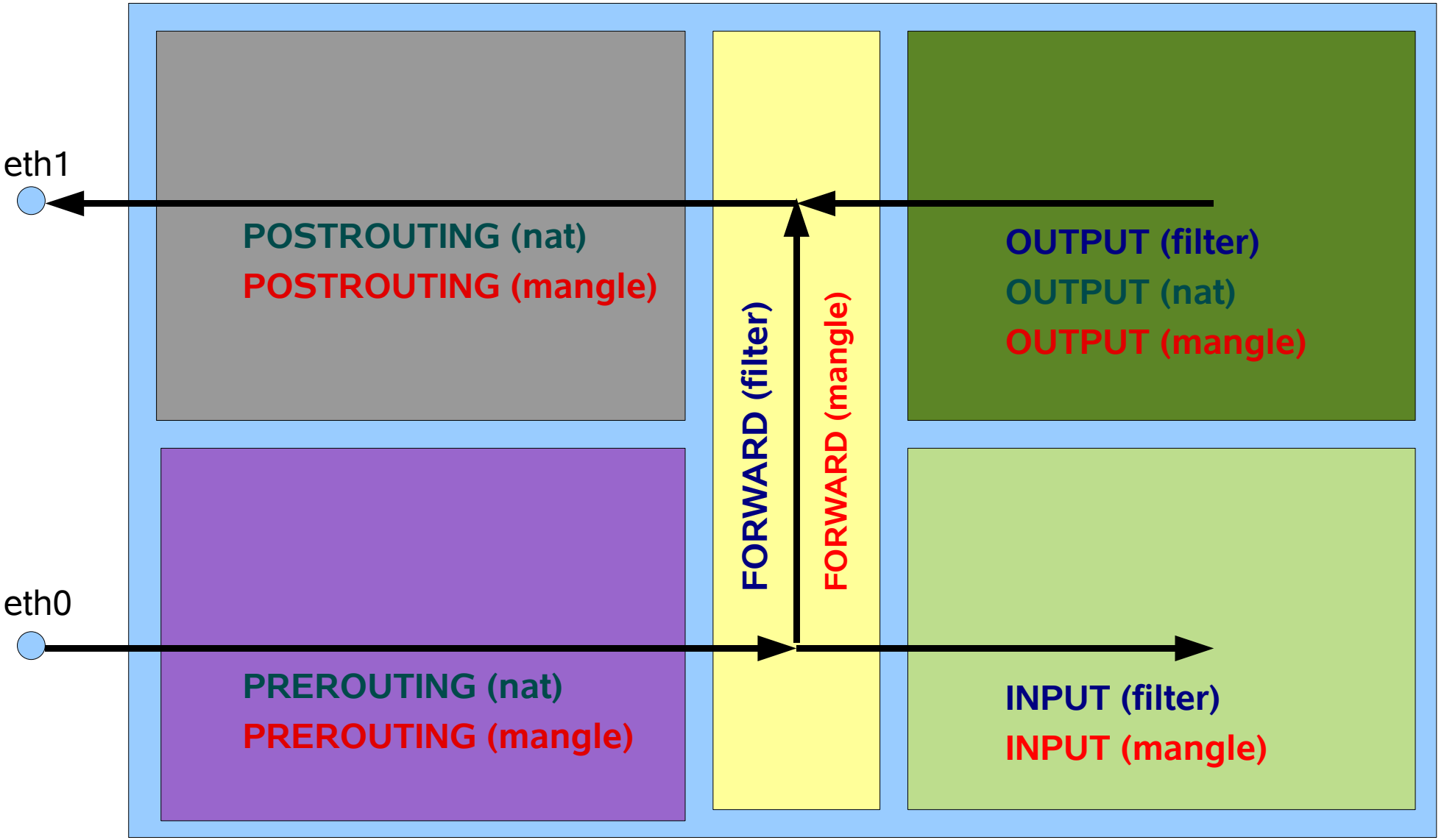
netfilter - пакетный фильтр Linux

Для управления пакетным фильтром используется программа iptables. Данная программа предназначена для определения правила фильтрации пакетов, выстраивая эти правила в цепочки, что позволяет задавать довольно сложную логику их обработки.

Кроме того, сами цепочки входят в состав следущих таблиц, каждая из которых предназначена для решения определенного класса задач:

- **filter** — фильтрация пакетов (таблица по умолчанию)
  - INPUT — цепочка для входящих пакетов
  - FORWARD — цепочка для транзитных пакетов
  - OUTPUT — цепочка для исходящих пакетов
- **nat** — Трансляция сетевых адресов
  - PREROUTING — цепочка DNAT-преобразований
  - POSTROUTING — цепочка SNAT-преобразований
  - OUTPUT — цепочка для исходящих пакетов
- **mangle** — модификация заголовков пакетов
  - PREROUTING — цепочка предварительных преобразований заловков пакетов
  - POSTROUTING — финальных преобразований заловков пакетов
  - INPUT — цепочка для входящих пакетов
  - FORWARD — цепочка для транзитных пакетов
  - OUTPUT — цепочка для исходящих пакетов

netfilter - пакетный фильтр Linux



iptables — управление пакетным фильтром

В общем виде правила записываются примерно так:

iptables [-t *table*] command [match] [target/jump]

Если в правило не включается спецификатор [-t table], то по умолчанию предполагается использование таблицы filter, если же предполагается использование другой таблицы, то это требуется указать явно.

Далее, непосредственно за именем таблицы, должна стоять команда. Если спецификатора таблицы нет, то команда всегда должна стоять первой. Команда определяет действие iptables, например: вставить правило, или добавить правило в конец цепочки, или удалить правило и т.п.

Раздел matches задает критерии проверки, по которым определяется подпадает ли пакет под действие этого правила или нет. Здесь мы можем указать самые разные критерии -- и IP-адрес источника пакета или сети, и сетевой интерфейс и т.д.

И наконец target указывает, какое действие должно быть выполнено при условии выполнения критериев в правиле. Здесь можно заставить ядро передать пакет в другую цепочку правил, "сбросить" пакет и забыть про него, выдать на источник сообщение об ошибке и т.п.

Таблица filter

Используется главным образом для фильтрации пакетов. Для примера, здесь мы можем выполнить **DROP**, **LOG**, **ACCEPT** или **REJECT** без каких либо сложностей, как в других таблицах. Имеется три встроенных цепочки:

- **FORWARD**, используемая для фильтрации пакетов, идущих транзитом через брандмауэр.
- **INPUT**, через эту цепочку проходят пакеты, которые предназначены локальным приложениям (брандмауэру).
- **OUTPUT** – используется для фильтрации исходящих пакетов, сгенерированных приложениями на самом брандмауэре.

Таблица nat

Используется главным образом для преобразования сетевых адресов (Network Address Translation). Через эту таблицу проходит только первый пакет из потока. Преобразования адресов автоматически применяется ко всем последующим пакетам. Это один из факторов, исходя из которых мы не должны осуществлять какую-либо фильтрацию в этой таблице. Она содержит следующие цепочки:

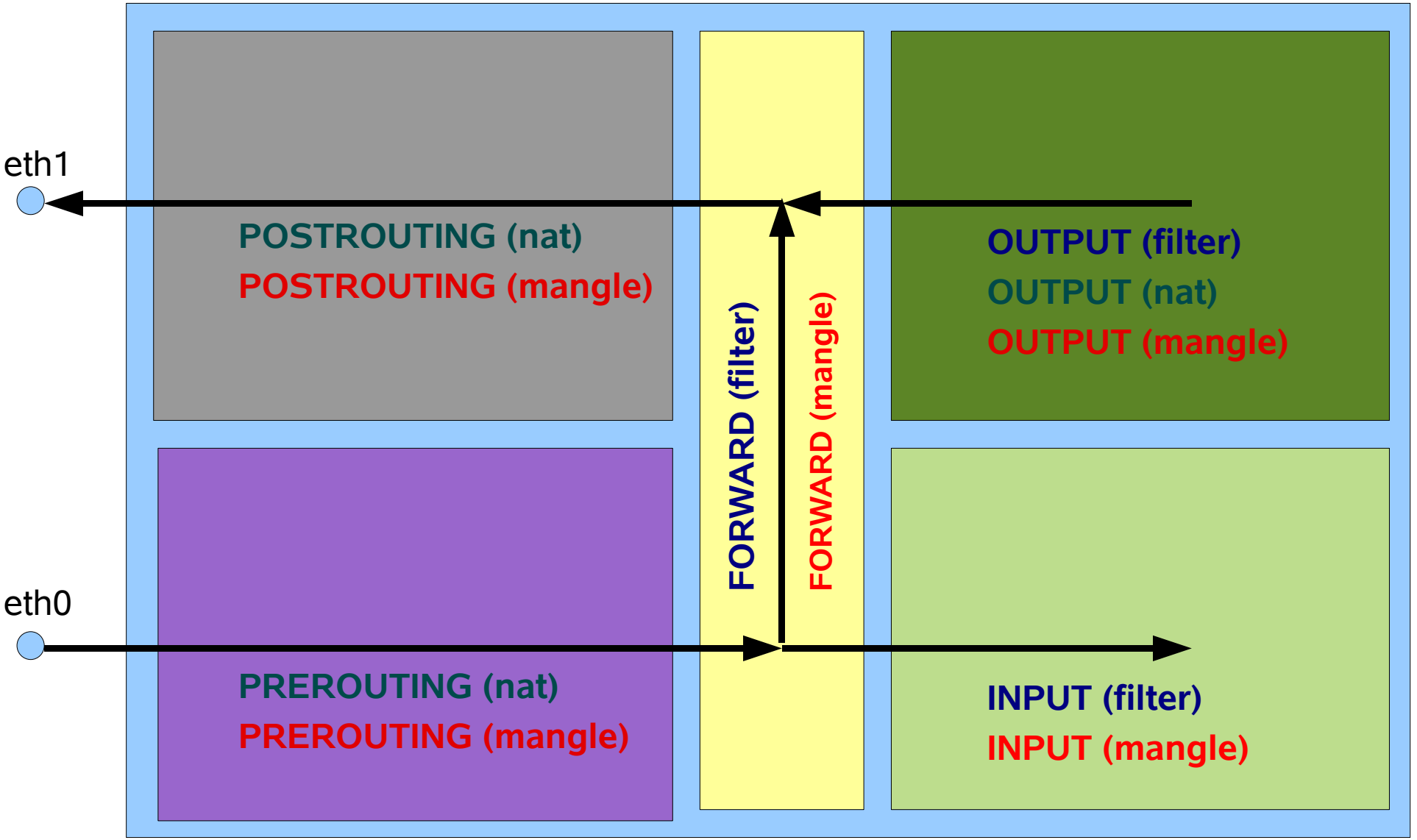
- **PREROUTING** – используется для внесения изменений в пакеты на входе в брандмауэр.
- **OUTPUT** – предназначена для преобразования пакетов, созданных приложениями внутри брандмауэра, перед принятием решения о маршрутизации.
- **POSTROUTING** – применяется для преобразования пакетов перед выдачей их во вне.

Таблица mangle

Эта таблица используется для внесения изменений в заголовки пакетов. Примером может служить изменение поля **TTL**, **TOS** или **MARK**. Важно: в действительности поле **MARK** не изменяется, но в памяти ядра заводится структура, которая сопровождает данный пакет все время его прохождения через машину, так что другие правила и приложения на данной машине (и только на данной машине) могут использовать это поле в своих целях. Заметьте, что таблица mangle ни в коем случае не должна использоваться для преобразования сетевых адресов или маскарадинга (Network Address Translation, Masquerading), поскольку для этих целей имеется таблица nat. Она содержит следующие цепочки:

- **PREROUTING** – используется для внесения изменений в пакеты на входе в брандмауэр.
- **POSTROUTING** – применяется для изменения заголовков пакетов перед выдачей их во вне.
- **INPUT**, через эту цепочку проходят пакеты, которые предназначены локальным приложениям (брандмауэру).
- **FORWARD**, используемая для модификации заголовков пакетов, идущих транзитом через брандмауэр.
- **OUTPUT** – для внесения изменений в заголовки пакетов, поступающих от приложений внутри брандмауэра.

Порядок прохождения пакетов



Команды iptables

-A – добавить правило в конец цепочки

-D – удалить правило из цепочки

-R – заменить одно правило другим

-I – вставить правило в указанное место в цепочке

-L – показать список правил

-F – очистить цепочку или таблицу

-Z – обнулить счетчики

-N – создать цепочку пользователя

-X – удалить цепочку пользователя

-P – установить политику по умолчанию

Команды iptables

-A – добавить правило в конец цепочки

При выполнении команды необходимо обязательно указать цепочку.

Пример:

iptables -A INPUT -s 10.10.103.100 -j ACCEPT

Команды iptables

-D – удалить правило из цепочки

Существуют два способа удаления правила: по его номеру или с указанием всех критериев отбора

Примеры:

iptables -D INPUT 1

iptables -D INPUT -p tcp --dport 80 -j DROP

Команды iptables

-R – заменить одно правило другим

Так как команда заменяет одно правило другим необходмо указать порядковый номер этого правила в цепочке.

Пример:

iptables -R INPUT 1 -s 192.168.0.1 -j ACCEPT

Команды iptables

-I – вставить правило в указанное место в цепочке

Команда вставляет правило под указанным номером и увеличивает на единицу номера всех последующих правил в этой цепочке

Пример:

iptables -I INPUT 1 -p tcp --sport 80 -j ACCEPT

Команды iptables

-L – показать список правил

Если не указывать имя цепочки, команда показывает все правила текущей таблицы. А если указать имя цепочки – показывет все правила текущей цепочки.

Пример:

iptables -L INPUT

Команды iptables

-F – очистить цепочку или таблицу

Если не указывать имя цепочки, команда удаляет все правила из текущей таблицы. А если указать имя цепочки – удаляет все правила из текущей цепочки.

Пример:

iptables -F INPUT

Команды iptables

-Z – обнулить счетчики

Каждому правилу в каждой из таблиц соответствуют два счетчика: количество пакетов и количество байт сработавших на данном правиле. Команда обнуляет счетчики в текущей таблице, или, если указано имя цепочки, в заданной цепочке.

Пример:

iptables -Z INPUT

Команды iptables

-N – создать цепочку пользователя

Команда создает цепочку с заданным именем в указанной таблице. Если таблица не указана, то подразумевается таблица filter. Имя создаваемой цепочки должно быть уникальным в пределах таблицы и не должно совпадать с зарезервированными именами цепочек и действий.

Примеры:

iptables -N tcp_filter

iptables -N udp_filter

iptables -N icmp_filter

Команды iptables

-X – удалить цепочку пользователя

Удаляет цепочку пользователя из указанной таблицы. Если таблица не указана, то подразумевается таблица filter. Удалять можно только цепочки не содержащие правил, кроме того в других цепочках не должно быть правил ссылающихся на данную цепочку.

Примеры:

iptables -X tcp_filter

iptables -X udp_filter

iptables -X icmp_filter

Команды iptables

-P – установить политику по умолчанию

Команда задает политику по умолчанию для указанной цепочки. Если таблица не указана, то подразумевается таблица filter. Политика определяет, что нужно сделать с пакетом, на котором не сработало ни одно правило данной цепочки. Разрешенные значения политики ACCEPT и DROP.

Примеры:

iptables -P INPUT DROP

iptables -P OUTPUT ACCEPT

Команды iptables

-A – добавить правило в конец цепочки

-D – удалить правило из цепочки

-R – заменить одно правило другим

-I – вставить правило в указанное место в цепочке

-L – показать список правил

-F – очистить цепочку или таблицу

-Z – обнулить счетчики

-N – создать цепочку пользователя

-X – удалить цепочку пользователя

-P – установить политику по умолчанию

Опции iptables

-v – показывать дополнительную информацию

-x – выводить точные значения счетчиков (без округления)

-n – не преобразовывать IP-адреса в DNS-имена

--line-numbers – включает вывод номеров правил в цепочке

-c – устанавливает значения счетчиков

Критерии отбора пакетов

Общие – не зависят от типа протокола и не требуют загрузки специальных модулей ядра

Неявные – зависят от типа протокола и не требуют загрузки специальных модулей ядра

Явные – требуют загрузки специальных модулей ядра

Общие критерии

-p – определяет протокол

-s – определяет IP-адрес источника

-d – определяет IP-адрес назначения

-i – определяет входящий интерфейс

-o – определяет исходящий интерфейс

-f – определяет фрагменты, фрагментированного пакета

Примеры:

-p tcp

-s 10.10.103.0/24

-o eth0

! -f

Внимание! Символ «!» инвертирует значение параметра.

Неявные критерии

TCP критерии:

--sport – порт источника

--dport – порт назначения

--tcp-flags – определение TCP-флагов

--syn – запрос на соединение

UDP критерии:

--sport – порт источника

--dport – порт назначения

ICMP критерии:

--icmp-type – определяет тип ICMP пакета

Примеры:

-dport 1024:65535

--icmp-type echo-request

Явные критерии

limit – ограничивает количество срабатываний правила

mac – позволяет использовать MAC-адреса в качестве критерия отбора пакетов

multiport – позволяет указать список портов

state – определяет состояния пакетов

Явные критерии. Критерий limit.

--limit-burst – определяет количество пакетов, по умолчанию значение критерия равно 5.

--limit – задает единицу времени в формате N/t

где

N – кол-во срабатываний в единицу времени

t – единица времени (s,m,h или d)

Данные критерии используются совместно для того, чтобы ограничить прохождение пакетов в единицу времени

Пример:

iptables -A INPUT -p icmp --icmp-type echo-request -m limit --limit 1/s \ --limit-burst 1 -j ACCEPT

Явные критерии. Критерий MAC.

--mac-source – Критерий определяет MAC-адрес сетевого узла, передавшего пакет. MAC-адрес должен указываться в формате XX:XX:XX:XX:XX:XX. Этот критерий имеет смысл только в цепочках PREROUTING, FORWARD и INPUT.

Символ «!» инвертирует значение параметра.

Пример:

iptables -A INPUT -m mac --mac-source 34:26:A6:1D:F7:04 -j ACCEPT

Явные критерии. Критерий multiport.

--source-port – Критерий используется для указания списка исходящих портов.

--destignation-port – Критерий используется для указания списка портов назначения.

Можно указать до 15 различных портов. Используется только с критериями -p tcp и -p udp.

Символ «!» инвертирует значение параметра.

Пример:

iptables -A INPUT -p tcp -m multiport --source-port 21,53 -j ACCEPT

Явные критерии. Критерий state.

--state – определяет состояние пакета.

Пакетный фильтр позволяет отслеживать не только TCP соединения, но и соединений по протоколам UDP и ICMP . Соединение может иметь одно из следующих состояний:

- **NEW** – пакет является первым для данного соединения
- **ESTABLISHED** – пакет принадлежит установленному соединению
- **RELATED** – пакет принадлежит к соединению связанному с уже установленным соединением
- **INVALID** – состояние пакета определить не удалось

Пример:

iptables -A INPUT -m state --state INVALID -j DROP

Действия и переходы

ACCEPT – принять пакет

DROP – сбросить пакет

REJECT – сбросить пакет с сообщением об ошибке

RETURN – возврат из цепочки

LOG – помещает информацию в системный журнал

Пример:

iptables -A INPUT -m state --state INVALID -j LOG \ --log-prefix «Strange:»

NAT преобразования

SNAT – замена IP-адреса или порта источника

MASQUERADE – является частным случаем SNAT-преобразования, который применяется в тех случаях когда IP-адрес получается динамически от DHCP-сервера.

DNAT – замена IP-адреса или порта назначения

REDIRECT – Выполняет перенаправление пакетов и потоков на другой порт той же самой машины.

SNAT

--to-source – определяет IP-адрес и порт на которые будут заменены соответствующие поля пакета.

Используется в тех случаях когда требуется организовать выход в интернет с компьютеров в локальной сети. Данное преобразование заменяет указанные в пакете IP-адрес и порт компьютера в локальной сети на IP-адрес и порт шлюза.

До SNAT		После SNAT	
IPs	Ports	IPs	Ports
192.168.1.2	6000	10.10.103.1	20001
192.168.1.3	6000	10.10.103.1	20002
192.168.1.4	6000	10.10.103.1	20003

Пример:

iptables -t nat -A POSTROUTING -o eth0 -j SNAT --to-source 10.10.103.X

MASQUERADE

--to-ports – определяет порт или диапазон портов

Используется в тех случаях когда требуется организовать выход в интернет с компьютеров в локальной сети. Данное преобразование заменяет указанные в пакете IP-адрес и порт компьютера в локальной сети на IP-адрес и порт шлюза.

Пример:

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

DNAT

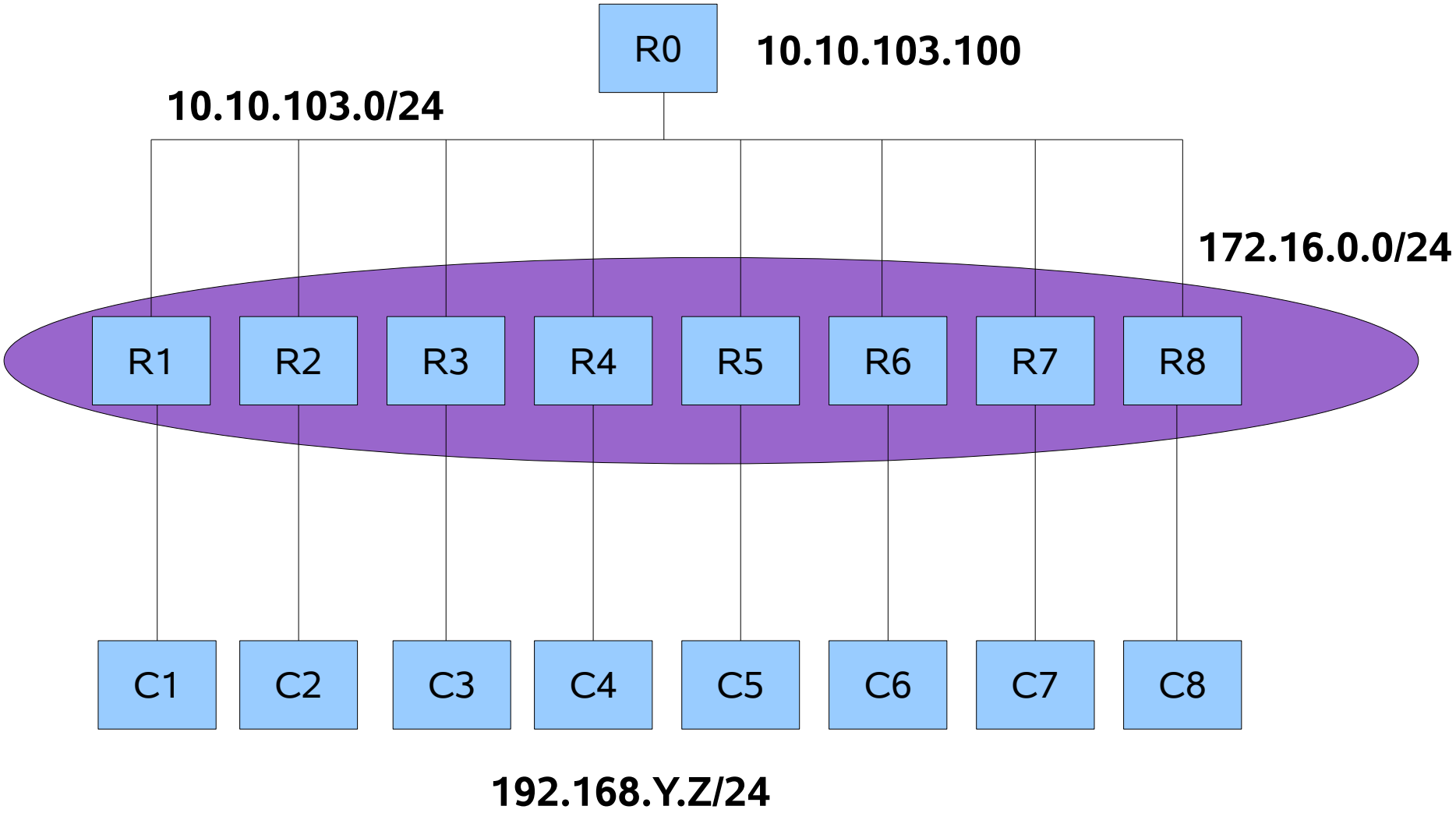
--to-destination – определяет IP-адрес и порт на которые будут заменены соответствующие поля пакета.

Используется в тех случаях когда требуется организовать доступ из интернета к компьютерам в DMZ. Данное преобразование заменяет указанные в пакете IP-адрес и порт компьютера в Интернет на IP-адрес и порт компьютера в локальной сети.

Пример:

iptables -t nat -A PREROUTING -i eth1 -j DNAT --to-destination 192.168.1.Y

Сегментная организация сети



Настройка роутера

```
#!/bin/bash
# Router configurator for 1-st of 8 network segments
echo 1 > /proc/sys/net/ipv4/ip_forward
ifconfig eth0 10.10.103.1 netmask 255.255.255.0 broadcast 10.10.103.255 up
ifconfig eth0:0 172.16.0.1 netmask 255.255.255.0 broadcast 172.16.0.255 up
ifconfig eth1 192.168.1.1 netmask 255.255.255.0 broadcast 192.168.1.255 up
route add default gw 10.10.103.100
route add -net  192.168.2.0/24 gw 172.16.0.2
route add -net  192.168.3.0/24 gw 172.16.0.3
route add -net  192.168.4.0/24 gw 172.16.0.4
route add -net  192.168.5.0/24 gw 172.16.0.5
route add -net  192.168.6.0/24 gw 172.16.0.6
route add -net  192.168.7.0/24 gw 172.16.0.7
route add -net  192.168.8.0/24 gw 172.16.0.8
iptables  -t  nat  -A   POSTROUTING  -o  eth0  -j  MASQUERADE
```

Настройка клиента

```
#!/bin/bash
# Client configurator for 1-st of 8 network segments
ifconfig eth0 down
ifconfig eth1 192.168.1.2 netmask 255.255.255.0 broadcast 192.168.1.255 up
route add default gw 192.168.1.1
```