



Сеть и сетевые сервисы в РЕД ОС

Имя сетевого интерфейса

Для смены имени интерфейса на любое другое можно переименовать в папке **/etc/sysconfig/network-scripts** переименовать файл **ifcfg-**<OLD>**** в файл **ifcfg-**<NEW>****, где **<OLD>** - старое название интерфейса, **<NEW>** - новое название интерфейса.

Пример статической настройки сетевого интерфейса:

```
TYPE="Ethernet"
```

```
ONBOOT="yes"
```

```
BOOTPROTO="static"
```

```
IPADDR="192.168.10.1"
```

```
NETMASK="255.255.255.0"
```

```
IPV4_FAILURE_FATAL="no"
```

```
IPV6INIT="no"
```

```
NAME=enp0s3
```

```
DEVICE="enp0s3"
```

Имя сетевого интерфейса

В новой версии NetworkManager конфигурационные файлы находятся в папке **/etc/NetworkManager/system-connections/**.

Теперь файлы имеют другую структуру:

[connection]

id=realme C21-Y

uuid=53659ae9-2adb-4dbc-b186-6e1f54efa42e

type=wifi

interface-name=wlp2s0

permissions=user:dima;

[wifi]

...

nmcli — Настройка Network Manager

nmcli [опции] объект [команда]

где основные объекты это

device - управление сетевыми интерфейсами;

connection - управление соединениями;

networking - управление сетью в целом;

general - отображение состояния сети и Network Manager;

radio - управление сетевыми протоколами, wifi, ethernet и т д.

Допустимые команды зависят от объекта

nmcli general status

nmcli connection show

Примеры использования nmcli

nmcli general status — просмотр состояния Network Manager

nmcli device status — просмотр состояния интерфейсов

nmcli connection show — просмотр доступных подключений

nmcli connection show eth0 — просмотр подробной информации об eth0

nmcli connection up eth0 - активация подключения по eth0

nmcli connection add con-name "dhcp" type ethernet ifname ens33 — создание подключения с именем «dhcp» типа ethernet для устройства ens33

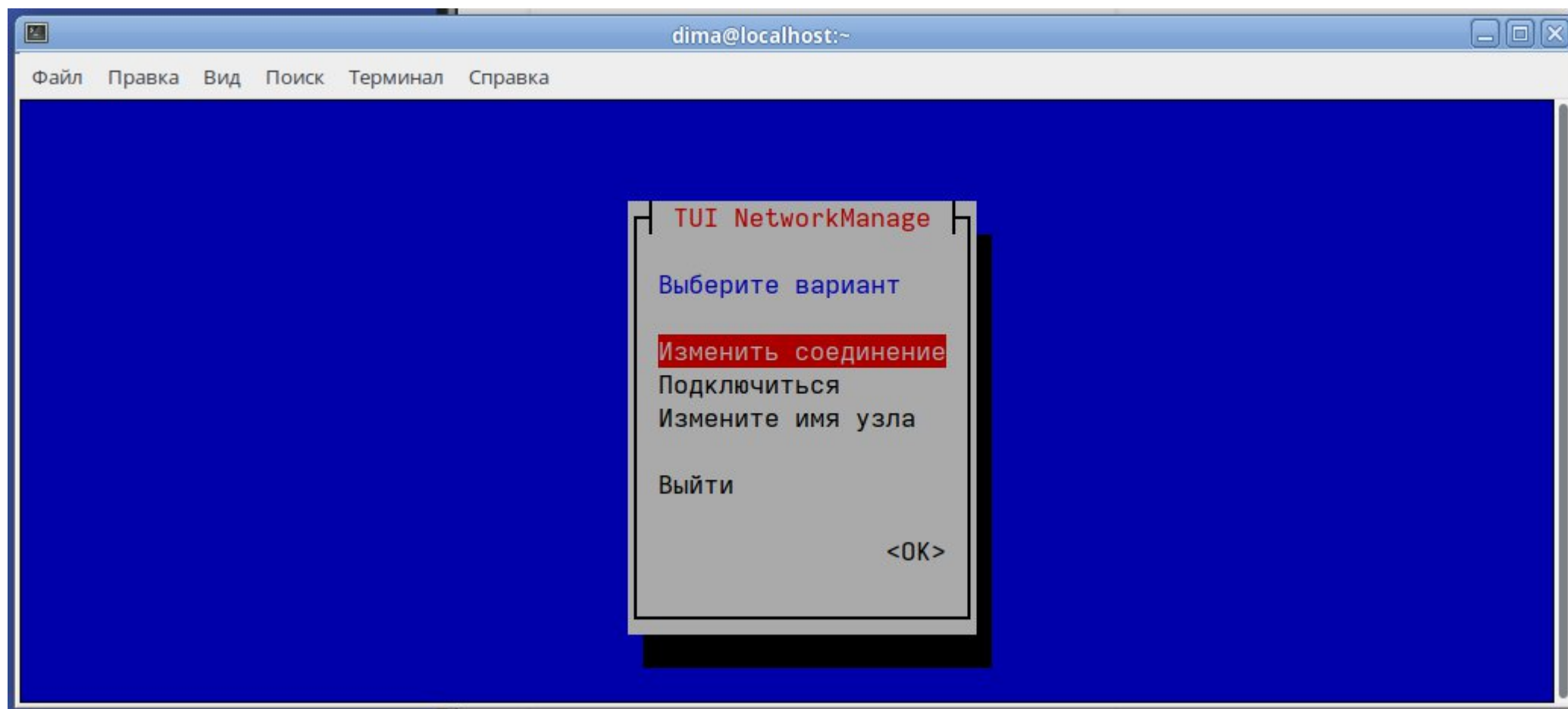
nmcli connection add con-name "static" ifname enp2s0 autoconnect no type ethernet ip4 192.168.0.210 gw4 192.168.0.1 — создание подключения со статическим IP-адресом

nmcli conn modify "static" ipv4.dns 8.8.8.8 — задание DNS-сервера

nmcli radio wifi on — включение WiFi

nmcli device wifi connect "TP-Link" password 12345678 name "TP-Link Wifi" — подключение к сети WiFi

Nmtui — настройка NetworkManager



ip - сетевые настройки

ip [опции] объект команда [параметры]

где объект может быть

address - сетевой адрес на устройстве

link - физическое сетевое устройство

monitor - мониторинг состояния устройств

neigh - ARP

route - управление маршрутизацией

rule - правила маршрутизации

tunnel - настройка туннелирования

Поддерживаются сокращения (address - a, link - l, route - r и т.д.)

ip - сетевые настройки

ip [опции] объект команда [параметры]

где опции

- v, **-Version** - только вывод информации об утилите и ее версии.
- h, **-human** - выводить данные в удобном для человека виде.
- s, **-stats** - включает вывод статистической информации.
- d, **-details** - показывать ещё больше подробностей.
- o, **-oneline** - выводить каждую запись с новой строки.
- r, **-resolve** - определять имена хостов с помощью DNS.
- a, **-all** - применить команду ко всем объектам.
- c, **-color** - позволяет настроить цветной, доступные значения: auto, always и never.
- br, **-brief** - выводить только базовую информацию для удобства чтения.

Примеры использования

ip link show - отобразить состояние всех сетевых интерфейсов

ip l sh - то же самое

ip l - то же самое

ip link show eth0 - отобразить состояние eth0

ip link set eth1 up - включить eth1

ip link set eth1 down - выключить eth1

ip address show - показать все ip адреса и их интерфейсы

ip address change 192.168.10.1/24 dev enp0s3 – смена адреса интерфейса

ip addr add 10.10.100.100/24 dev enp3s0 – добавление ip адреса

ip a l permanent - отобразить только статические ip адреса

ip a l dynamic - отобразить только динамические ip адреса

ip addr add 1.1.1.13/24 dev eth0 - установить ip адрес для интерфейса eth0

ip addr del 1.1.1.13/24 dev eth0 - удалить ip адрес интерфейса eth0

Примеры использования

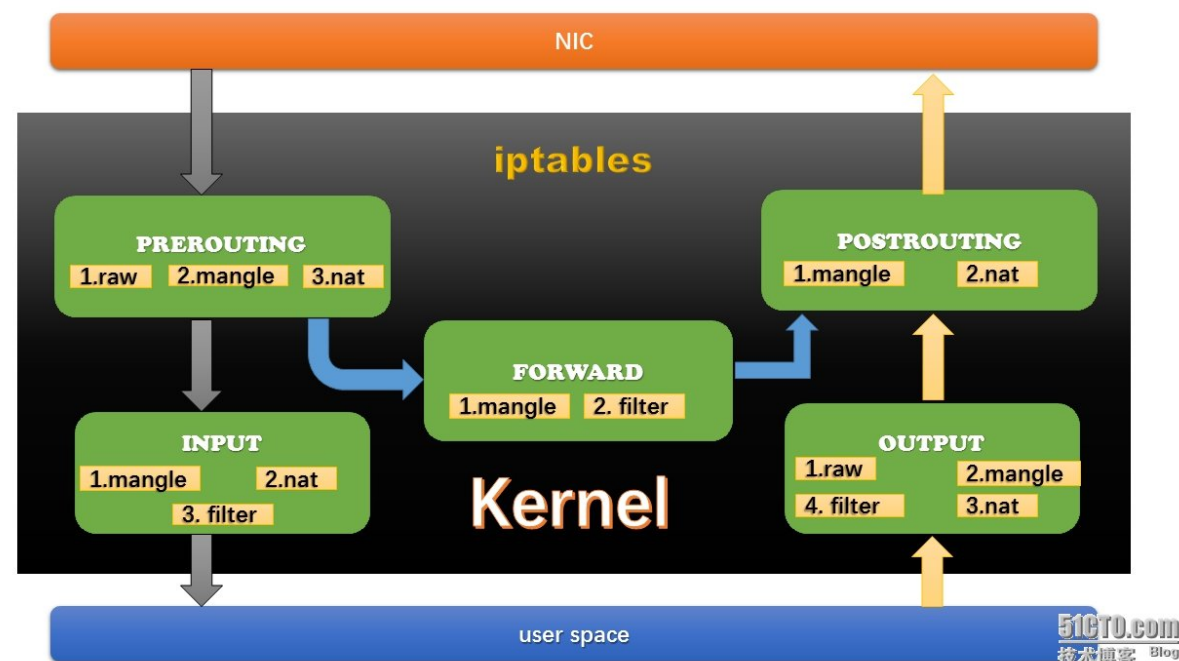
ip link set dev enp0s3 address AA:BB:CC:DD:EE:FF – смена MAC
ip r sh - показать все маршруты в таблице маршрутизации
ip route get 10.10.20.0/24 - отобразить маршрут к этой сети
ip route add 10.10.20.0/24 via 192.168.50.100 - создать маршрут
ip route delete 10.10.20.0/24 - удалить маршрут.
ip neigh show dev eth0 - посмотреть все ARP записи для eth0
ip neigh del dev enp0s3 192.168.0.105 – удаление записи ARP
ip neigh flush dev enp0s3 – удаление ARP записи интернета
ip neigh flush – очистка ARP записей

Подсистема iptables и netfilter

Подсистема iptables и netfilter уже достаточно давно встроена в ядро Linux. Там эти пакеты подвергаются проверкам и затем для каждой проверки, если она пройдена, выполняется указанное в ней действие.

Все пакеты делятся на три типа: входящие (**INPUT**), исходящие (**OUTPUT**) и проходящие (**FORWARD**).

Над цепочками правил в iptables есть ещё один уровень абстракции — **таблицы**, — которые предназначены для выполнения разных действий над пакетами, например для модификации или фильтрации



Подсистема iptables и netfilter

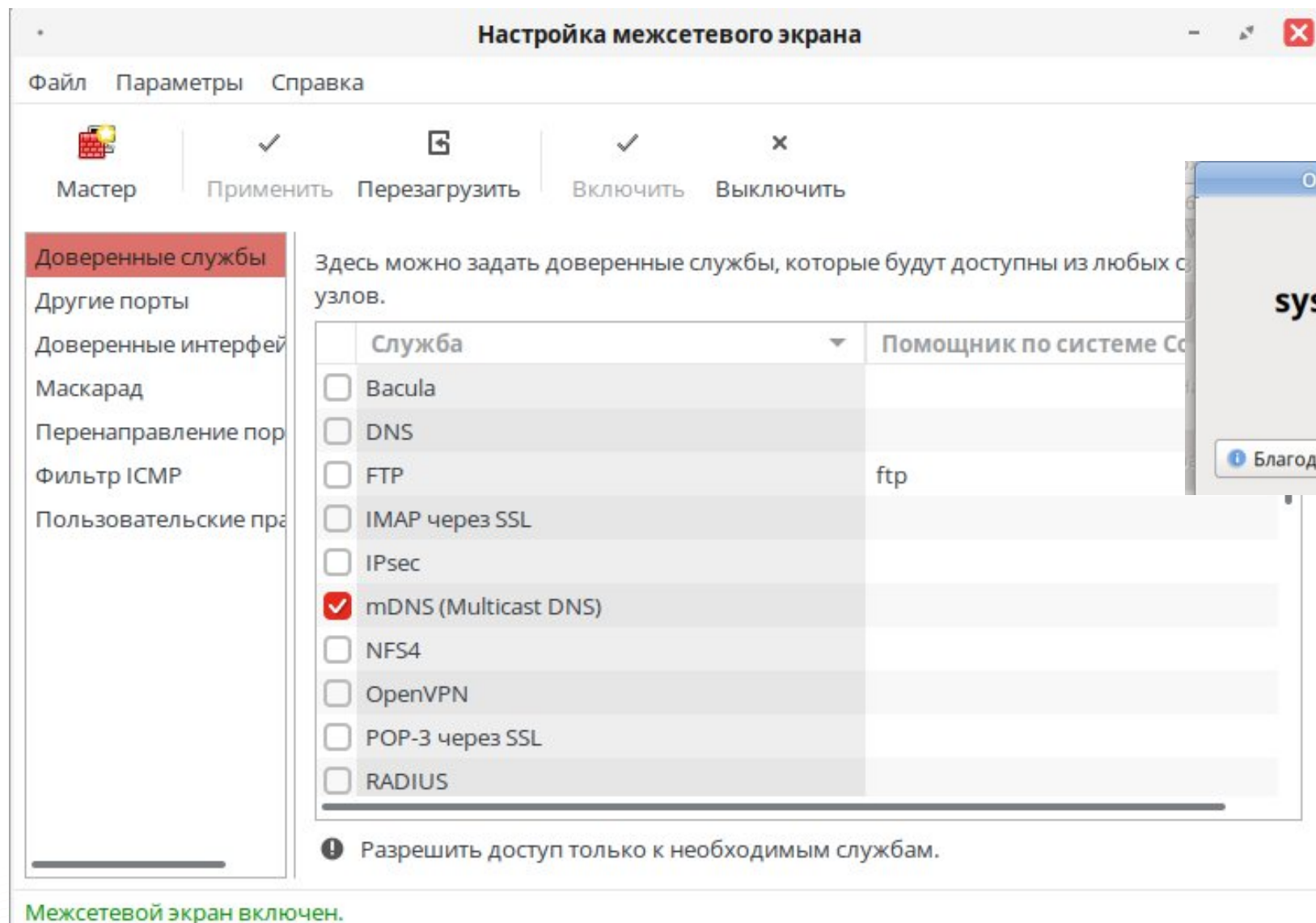
Для каждого типа пакетов можно установить набор правил, которые по очереди будут проверяться на соответствие с пакетом и если пакет соответствует, то применять к нему указанное в правиле действие. Правила образуют цепочку. Действий может быть несколько, например:

ACCEPT — разрешить прохождение пакета дальше по цепочке правил;

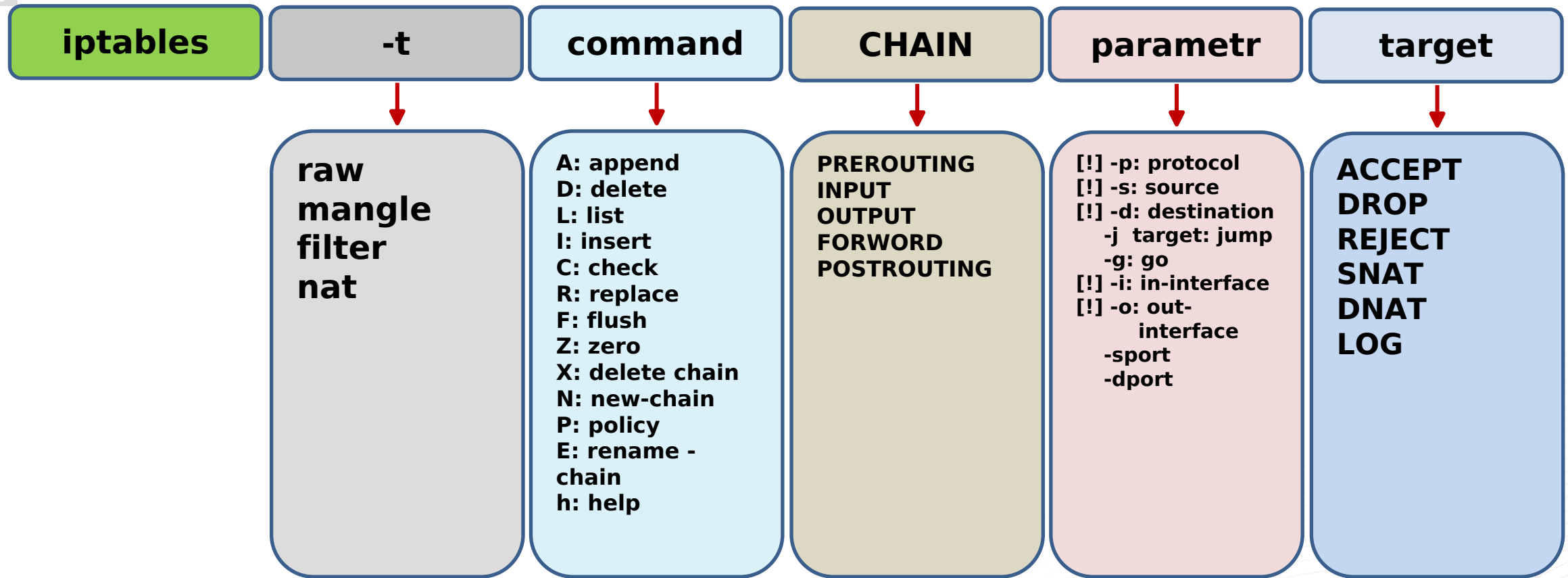
DROP — удалить пакет;

REJECT — отклонить пакет, отправителю будет отправлено сообщение об этом.

Подсистема iptables и netfilter



Команда iptables



Подсистема iptables и netfilter

Подсистема **iptables** и **netfilter** встроены в ядро. Набор утилит (пакет iptables) устанавливается в системе по умолчанию.

Для управления цепочками используется утилита iptables, общий вид которой **sudo iptables -L** таблица действие цепочка парам.

Например, запретить все входящие с адреса 192.168.10.2:

```
iptables -A INPUT -s 192.168.10.2 -j DROP
```

Разрешаем все пакеты со статусом установленные:

```
Iptables -A INPUT -m state --tate RELATED,ESTABLISHED -j ACCEPT
```

Подсистема iptables и netfilter

Разрешаем ssh

```
# iptables -A INPUT -p tcp --dport ssh -j ACCEPT
```

Делаем запрещающую политику входных правил

```
iptables -P INPUT DROP
```

Разрешаем пинги ПК

```
iptables -A INPUT -p icmp -j ACCEPT
```

Подсистема iptables и netfilter

Очистка правил

```
sudo iptables -F
```

Или очистка конкретного правила

```
sudo iptables -D INPUT -s 192.168.10.2 -j DROP
```

Сохраняем правила в /etc/sysconfig/iptables

```
sudo /sbin/service iptables save
```

Подсистема iptables и netfilter

Включение функции маскарадинга - это подмена некоторых параметров в заголовках IP пакетов, позволяющая машинам, не имеющим реальных IP адресов полноценно работать в Интернет.

с помощью iptables

```
iptables -t nat -A POSTROUTING -o <IFACE> -j MASQUERAD
```

или с помощью firewallld

```
firewall-cmd --permanent --direct --passthrough ipv4 -t nat -I POSTROUTING -o  
<IFACE> -j MASQUERADE -s 192.168.1.0/24
```

где **<IFACE>** — имя сетевого интерфейса глобальной сети
192.168.1.0 — адрес внутренней сети

Подсистема iptables и netfilter

Таблица маршрутизации задаёт правила доставки пакетов до места назначения

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
0.0.0.0	185.61.25.129	0.0.0.0	UG	0	0	0	enp3s0
10.0.6.0	10.81.254.1	255.255.255.252	UG	0	0	0	tun0
10.0.24.0	10.81.254.1	255.255.248.0	UG	0	0	0	tun0
10.81.0.0	10.81.254.1	255.255.255.0	UG	0	0	0	tun0
10.81.1.0	0.0.0.0	255.255.255.0	U	0	0	0	enp5s0f1.1
10.81.2.0	10.81.10.2	255.255.255.0	UG	0	0	0	tun1

- Destination – IP-адрес назначения;
- Gateway – шлюз, на который нужно отправить пакет;
- Genmask – маска подсети;
- Flags – флаги маршрутизации;
- Metric – «расстояние» до цели, в последних ядрах не используется;
- Ref – количество ссылок на это правило (не используется в Linux);
- Use – количество использований этого правила;
- Iface – сетевой интерфейс, через который должен идти пакет

Подсистема iptables и netfilter

Управлять текущими таблицами маршрутизации можно с помощью утилит route или ip. Например:

- просмотр таблиц маршрутизации route или ip route
- добавление статического маршрута

```
route add -net 192.168.1.0 255.255.255.0 gw 192.168.0.1
```

или

```
ip route add 192.168.1.0/24 via 192.168.0.1
```

В случае постоянного использования статического маршрута необходимо сохранить нужное правило маршрутизации в файле

```
/etc/sysconfig/network-scripts/route-<net-device>
```


tcpdump

Каждая строка включает:

Метка времени Unix (**20: 58: 26.765637**)

протокол (**IP**)

имена или IP-адреса и номер порта (**10.0.0.50.80**)

Флаги TCP (**Flags [F.]**). Указывают на состояние соединения

Порядковый номер данных в пакете. (**seq 1**)

Номер подтверждения. (**ack 2**)

Размер окна (**win 453**). Количество байтов. Далее параметры TCP

Длина полезной нагрузки данных. (**length 0**)

Wireshark — программа-анализатор трафика для компьютерных сетей Ethernet. Wireshark умеет работать с множеством форматов входных данных, соответственно, можно открывать файлы данных, захваченных другими программами.



Команда nmap

Сканер сети. Позволяет понять какие компьютеры подключены к сети, узнать их имена, а также посмотреть какое программное обеспечение на них установлено, какая операционная система и какие типы фильтров применяются. **Функциональность** программы может быть **расширена** за счет собственного **скриптового** языка.

nmap опции адрес

Сканирование диапазона адресов

nmap -sn 192.168.10.1/24

или

nmap -sL 192.168.10.1/24

Команда nmap

Сканирование открытых портов

```
nmap -sV 192.168.0.8
```

Сканирование доступного порта

```
nmap -sC 192.168.56.102 -p 21
```

Посмотреть все скрипты используемые программой

```
find /usr/share/nmap/scripts/ -name '*.nse'
```

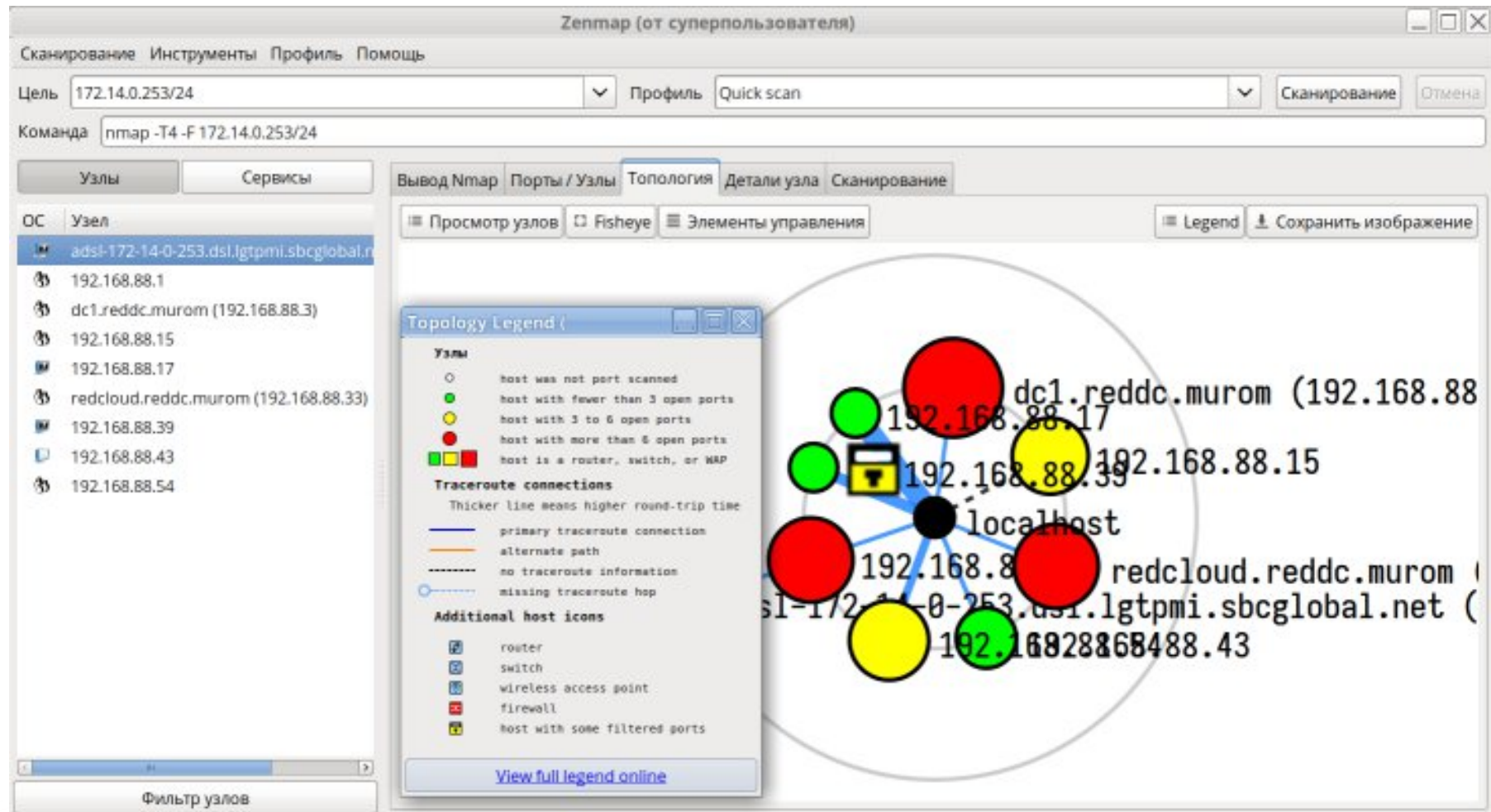
Просмотр описания скрипта

```
nmap --script-help http-passwd.nse
```

Запуск скриптов для smb

```
nmap --script smb-*
```

zenmap – это официальная версия Nmap с графическим интерфейсом пользователя (GUI).





Спасибо за внимание!

www.red-soft.ru
redos@red-soft.ru

