ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 10. «МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО КОМАНДАМ УПРАВЛЕНИЯ СЕТЬЮ В UNIX»

Теоретическая часть

Команды по конфигурированию сети

Для настройки сетевых интерфейсов используется команда ifconfig, имеющая следующий синтаксис:

ifconfig [-L] [-m] interface [create] [address_family] [address [dest_address]] [parameters] ifconfig interface destroy ifconfig -a [-L] [-d] [-m] [-u] [address_family] ifconfig -l [-d] [-u] [address_family] ifconfig [-L] [-d] [-m] [-u] [-C]

Команда может использоваться при загрузке системы для настройки адресов каждого сетевого интерфейса, а также после загрузки для изменения параметров сетевых интерфейсов. Если команда введена без аргументов, **ifconfig** выдает информацию о состоянии активных интерфейсов. Если в качестве аргумента указан какой-либо интерфейс, то выдается информация только о состоянии этого интерфейса; если указан один аргумент -a, выдается информация о состоянии всех интерфесов, даже отключенных. Пример:

```
user@desktop$ ifconfig rl0
rl0:
flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST>  mtu
1500
options=8<VLAN_MTU>
inet6 fe80::250:22ff:febb:5f1%rl0  prefixlen  64 scopeid  0x3
inet  192.168.19.86  netmask  0xffffff00  broadcast
192.168.19.255
ether 00:50:22:bb:05:f1
media: Ethernet autoselect (100baseTX <full-duplex>)
status: active
```

Иначе команда конфигурирует указанный интерфейс. Изменить настройки какого-либо интерфейса может только суперпользователь.

Опции:

интерфейс	– имя интерфейса (например, rl0 в BSD или eth0 в Linux).
up	– вызывает активизацию интерфейса. Задается неявно при присвоении адреса интерфейсу.
down	– вызывает остановку работы драйвера для интерфейса.
[-]arp	– включает или отключает использование протокола ARP для интерфейса.
[-]promisc	– включает или отключает неразборчивый режим (promiscuous mode) работы интерфейса. В этом режиме все проходящие по сети пакеты будут приниматься интерфейсом.
[-]allmulti	– включает или отключает режим all-multicast. В этом режиме все многоадресные (multicast) пакеты в сети будут приниматься интерфейсом.
metric N	– устанавливает метрику интерфейса.
mtu N	– устанавливает максимальный размер пакета (Maximum Transfer Unit - MTU) для интерфейса.
адрес	– IP-адрес, присваиваемый интерфейсу.
netmask адрес	– устанавливает маску сети IP для этого интерфейса. По умолчанию используется обычная маска сети класса A, B или C (что определяется по IP-адресу интерфейса), но можно усановить любое значение.
add адрес/длина_префикса	– добавляет адрес IPv6 для интерфейса.

del адрес/длина_префикса	– удаляет адрес IPv6 для интерфейса.
irq адрес	– устанавливает аппаратное прерывание, используемое устройством. Не для всех устройств можно динамически менять значение IRQ.
media тип	– устанавливает физический порт или тип носителя, используемый устройством. Не для всех устройств можно менять этот параметр, и для разных устройств могут поддерживаться различные значения. Типичные значения типа - 10base2 (коаксиальный кабель Ethernet), 10baseT (витая пара Ethernet 10 Мбит/сек), AUI (внешний передатчик) и т.д. Специальный тип носителя auto можно использовать, чтобы потребовать от драйвера автоматически обпределять тип носителя. Не все драйверы могут это делать.
[-]broadcast [адрес]	– если указан аргумент адрес, задает соответствующий протоколу широковещательный адрес для интерфейса. В противном случае устанавливает (или сбрасывает) флаг IFF_BROADCAST для интерфейса.

Пример изменения IP-адреса интерфейса rl0:

```
user@desktop ~ $ ifconfig rl0
rl0:
flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST>  mtu
1500
        options=8<VLAN_MTU>
        inet6 fe80::250:22ff:febb:5f1%rl0 prefixlen 64 scopeid
0x3
        inet 192.168.19.86 netmask 0xffffff00 broadcast
192.168.19.255
        ether 00:50:22:bb:05:f1
```

```
        media: Ethernet autoselect (100baseTX <full-duplex>)
        status: active
user@desktop ~ $ ifconfig rl0 192.168.0.1
user@desktop ~ $ ifconfig rl0
rl0:
flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST>  mtu
1500
        options=8<VLAN_MTU>
        inet6 fe80::250:22ff:febb:5f1%rl0 prefixlen 64 scopeid
0x3
        inet 192.168.0.1 netmask 0xffffff00 broadcast
192.168.19.255
        ether 00:50:22:bb:05:f1
        media: Ethernet autoselect (100baseTX <full-duplex>)
        status: active
```

Команда **arp** отображает ARP-таблицу данного хоста. с помощью параметра *-i* можно специфицировать сетевой интерфейс, информация о котором интересует.

```
desktop ~ # arp -i eth0
Address                  HWtype  HWaddress           Flags Mask
Iface
DIMON.mshome.net         ether   00:50:BF:12:8A:9E   C
eth0
```

Таблица с информацией о канальном уровне содержит связь IP- и MAC-адресов. При использовании параметра *-n* IP-адреса не будут заменяться символьными именами хостов.

Команда **route** используется для просмотра и изменения таблицы маршрутизации хоста. Для этой команды также работает параметр *-n*, при использовании которого IP-адреса не будут заменяться символьными именами хостов.

Пример обычной таблицы маршрутизации для отдельного компьютера в сети:

```
desktop ~ # route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref
Use Iface
192.168.5.0     0.0.0.0         255.255.255.0   U     0      0
0 eth1
127.0.0.0       0.0.0.0         255.0.0.0       U     0      0
0 lo
0.0.0.0         192.168.5.254   0.0.0.0         UG    0      0
0
```

eth1

Особый интерес представляет адрес 0.0.0.0, который соответствует хосту назначения по умолчанию.

Для добавление нового маршрута к определённому хосту используются параметры *add* и *-host*:

desktop ~ # route add -host 192.168.0.1 eth0

Эта команда создаёт новую строку в таблице маршрутизации, согласно которой все пакеты к узлу 192.168.0.1 должны отправляться в сетевой интерфейс eth0.

Также можно добавлять шлюз для отправки пакетов в определённую сеть или к хосту:

desktop ~ # route add -net 192.168.1.0 gw 192.168.0.5

Таким образом, все пакеты для сети 192.168.1.0 будут направляться на узел 192.168.0.5.

Аналогично, маршруты удаляются параметром *del* с указанием всей информации о маршруте:

desktop ~ # route del default gw 192.168.0.1

Эта команда удаляет маршрут по умолчанию через хост 192.168.0.1.

Команды по диагностике сети

Для посылки пакетов ICMP ECHO_REQUEST сетевым хостам используется команда **ping,** имеющая следующий синтаксис:

ping [-AaDdfnoQqRrv] [-c число_пакетов] [-i секунд] [-l preload] [-M mask | time] [-m ttl] [-P policy] [-p pattern] [-S src_addr] [-s packetsize] [-t timeout] [-z tos] host ping [-AaDdfLnoQqRrv] [-c число_пакетов] [-I iface] [-i секунд] [-l preload] [-M mask | time] [-m ttl] [-P policy] [-p pattern] [-S src_addr] [-s packetsize] [-T ttl] [-t timeout] [-z tos] mcast-group

Команда **ping** использует датаграмму ECHO_REQUEST протокола ICMP, чтобы вызвать ответ ICMP ECHO_RESPONSE указанного хоста или сетевого шлюза. Если хост отвечает, **ping** выдает сообщение, что хост включен (хост is alive), в стандартный выходной поток.

Для проверки наличия хоста в сети достаточно ввести команду **ping** с аргументом - именем или адресом хоста:

user@desktop$ ping yandex.ru

64 bytes from 213.180.204.11: icmp_seq=0 ttl=48 time=5.659 ms

64 bytes from 213.180.204.11: icmp_seq=1 ttl=48 time=5.404 ms

64 bytes from 213.180.204.11: icmp_seq=2 ttl=48 time=4.889

ms

^C

--- yandex.ru ping statistics ---

3 packets transmitted, 3 packets received, 0% packet loss

round-trip min/avg/max/stddev = 4.889/5.317/5.659/0.320 ms

Для отправки определенного числа пакетов необходимо указать опцию -c <число_пакетов>. Для установки интервала между отправкой пакетов используется опция -i <количество секунд>.

Для отладки сетевых соединений посредством построения маршрута следования пакетов к хосту назначения служит команда **traceroute**. Для этой команды также работает параметр -n, при использовании которого IP-адреса не будут заменяться символьными именами хостов.

Пример следования пакетов до хоста ya.ru:

desktop ~ # traceroute ya.ru

traceroute to ya.ru (213.180.204.8), 64 hops max, 40 byte packets

1 195.91.230.65 (195.91.230.65) 0.890 ms 1.907 ms 0.809 ms

2 cs7206.rinet.ru (195.54.192.28) 0.895 ms 0.769 ms 0.605 ms

3 ix2-m9.yandex.net (193.232.244.93) 1.855 ms 1.519 ms 2.95 ms

4 c3-vlan4.yandex.net (213.180.210.146) 3.412 ms 2.698 ms 2.654 ms

5 ya.ru (213.180.204.8) 2.336 ms 2.612 ms 3.482 ms.

Команда **netstat** используется для показа состояния сети и имеет следующий синтаксис:

netstat [-AaLnSW] [-f protocol_family | -p protocol] [-M core] [-N system]

Команда **netstat** показывает содержимое различных структур данных, связанных с сетью, в различных форматах в зависимости от указанных опций. Первая форма команды показывает список активных сокетов (sockets) для каждого протокола. Вторая форма выбирает одну из нескольких других сетевых структур данных. Третья форма показывает динамическую статистику пересылки пакетов по сконфигурированным сетевым интерфейсам; аргумент интервал задает, сколько секунд собирается информация между последовательными показами.

Опции:

-a	– показывать состояние всех сокетов; обычно сокеты, используемые серверными процессами, не показываются.
-A	– показывать адреса любых управляющих блоков протокола, связанных с сокетами; используется для отладки.
-i	– показывать состояние автоматически сконфигурированных (auto-configured) интерфейсов. Интерфейсы, статически сконфигурированные в системе, но не найденные во время загрузки, не показываются.
-n	– показывать сетевые адреса как числа. netstat обычно показывает адреса как символы. Эту опцию можно использовать с любым форматом показа.
-r	– показать таблицы маршрутизации. При использовании с опцией -s, показывает статистику маршрутизации.
-s	– показать статистическую информацию по протоколам. При использовании с опцией -r, показывает статистику маршрутизации.
-f семейство_адресов	– ограничить показ статистики или адресов управляющих блоков только указанным семейством_адресов, в качестве которого можно указывать: net Для семейства адресов AF_INET nix Для семейства адресов AF_UNIX

-I интерфейс	– выделить информацию об указанном интерфейсе в отдельный столбец; по умолчанию (для третьей формы команды) используется интерфейс с наибольшим объемом переданной информации с момента последней перезагрузки системы. В качестве интерфейса можно указывать любой из интерфейсов, перечисленных в файле конфигурации системы, например, emd1 или lo0.
-p имя_протокола	– Ограничить показ статистики или адресов управляющих блоков только протоколом с указанным именем_протокола, например, tcp.

Пример показа таблицы маршрутизации:

```
user@desktop ~$ netstat -r
Routing tables
Internet:
Destination        Gateway            Flags     Refs      Use   Netif
Expire
default            19-101.local       UGS          0  1373769     rl0
localhost          localhost          UH           1      290     lo0
192.168.0          link#1             UC           0        0     dc0
192.168.19         link#3             UC           0        0     rl0
19-86.local        localhost          UGHS         0        0     lo0
19-101.local       00:0d:bc:e4:27:bf  UHLW         1        0     rl0
116

Internet6:
Destination        Gateway            Flags      Netif  Expire
localhost.prov.ru  localhost.prov.ru  UH           lo0
fe80::%dc0         link#1             UC           dc0
fe80::2a0:ccff:fe3 00:a0:cc:3d:1f:bd  UHL          lo0
fe80::%rl0         link#3             UC           rl0
fe80::250:22ff:feb 00:50:22:bb:05:f1  UHL          lo0
fe80::%lo0         fe80::1%lo0        U            lo0
fe80::1%lo0        link#5             UHL          lo0
ff01::             localhost.prov.ru  U            lo0
ff02::%dc0         link#1             UC           dc0
```

```
ff02::%rl0        link#3            UC          rl0
ff02::%lo0        localhost.prov.ru UC          lo0
```

Команда **host** служит для получения доменной информации о хосте: IP-адреса, MX-записи и другой информации, связанной с данным символьным именем. Имя хоста указывается в качестве аргумента команды.

Пример работы команды:

```
user@desktop ~$ host yandex.ru
yandex.ru has address 213.180.204.11
yandex.ru mail is handled by 10 mx2.yandex.ru.
yandex.ru mail is handled by 0 mx1.yandex.ru.
```

Вторым аргументом можно указать DNS-сервер, который будет использоваться при получении этой информации:

```
user@desktop ~$ host yandex.ru ns1.aiya.ru
Using domain server:
Name: ns1.aiya.ru
Address: 85.142.20.152#53
Aliases:

yandex.ru has address 213.180.204.11
Using domain server:
Name: ns1.aiya.ru
Address: 85.142.20.152#53
Aliases:

Using domain server:
Name: ns1.aiya.ru
Address: 85.142.20.152#53
Aliases:

yandex.ru mail is handled by 0 mx1.yandex.ru.
yandex.ru mail is handled by 10 mx2.yandex.ru.
```

Команда **tcpdump** используется для мониторинга сети на канальном и более высоких уровнях. Программа «слушает» на одним или нескольких сетевых интерфейсах и выводит дамп пакетов, проходящих через этот интерфейс.

Параметр *-i* задаёт имя сетевого интерфейса, на котором запускается прослушивание. При просмотре захватываемых данных удобно использовать ключ *-l*, который буферизует вывод построчно. Для этой команды также работает параметр *-n*, при ис-

пользовании которого IP-адреса не будут заменяться символьными именами хостов.

Пример работы команды:

desktop ~ # tcpdump -i eth0 -l -n

tcpdump: verbose output suppressed, use -v or -vv for full protocol decode

listening on eth1, link-type EN10MB (Ethernet), capture size 96 bytes

12:51:07.486755 arp who-has 0.0.0.0 (00:30:48:2b:6d:6a) tell 0.0.0.0

12:51:12.486606 arp who-has 0.0.0.0 (00:30:48:2b:6d:6a) tell 0.0.0.0

12:51:14.457608 IP 192.168.5.23.56385 > 194.91.250.11.443: P 3645922938:3645923156(218) ack 2092518729 win 10086

12:51:14.491343 IP 194.91.250.11.443 > 192.168.5.23.56385: . ack 218 win 10720

Для вывода расширенной информации о пакетах используются ключи *-v* или *-vv*.

desktop ~ # tcpdump -i eth1 -l -n -vv

tcpdump: listening on eth1, link-type EN10MB (Ethernet), capture size 96 bytes

12:57:53.043797 IP (tos 0x0, ttl 51, id 46031, offset 0, flags [DF], proto: TCP (6),

length: 286) 194.91.250.11.5190 > 192.168.5.23.38993: P 2517343058:2517343292(234)

ack 2346573376 win 2202 <nop,nop,timestamp 2713588760 497668>

12:57:53.043865 IP (tos 0x0, ttl 64, id 52382, offset 0, flags [DF], proto: TCP (6),

length: 52) 192.168.5.23.38993 > 194.91.250.11.5190: ., cksum 0x1fd7 (correct),

1:1(0) ack 234 win 11945 <nop,nop,timestamp 506366 2713588760>

12:57:53.401516 IP (tos 0x0, ttl 48, id 45237, offset 0, flags [DF], proto: TCP (6),

length: 210) 194.91.250.11.443 > 192.168.5.23.56385: P 2092522043:2092522213(170)

ack 3645927446 win 10720

...

Команда **tcpdump** обладает очень богатым интерфейсом, включающим условные выражения, по которым должны выделяться интересующие пакеты. Например, можно использовать

условия удалённого порта (равно 80):

desktop ~ # tcpdump -i eth1 -l -n -vv dst port 80

tcpdump: listening on eth1, link-type EN10MB (Ethernet), capture size 96 bytes

13:55:36.563959 IP (tos 0x0, ttl 64, id 3936, offset 0, flags [DF], proto: TCP (6),

length: 60) 192.168.5.23.52348 > 213.180.204.11.80: S, cksum 0x2766 (correct),

3855548287:3855548287(0) win 5840 <mss 1460,sackOK,timestamp 1372191 0,nop,wscale 2>

13:55:36.592654 IP (tos 0x0, ttl 64, id 3937, offset 0, flags [DF], proto: TCP (6),

length: 40) 192.168.5.23.52348 > 213.180.204.11.80: ., cksum 0xebc5 (correct),

3855548288:3855548288(0) ack 3869420799 win 5840

13:55:36.592731 IP (tos 0x0, ttl 64, id 3938, offset 0, flags [DF], proto: TCP (6),

length: 627) 192.168.5.23.52348 > 213.180.204.11.80: P 0:587(587) ack 1 win 5840

Команда **nmap** – сетевой сканер, с помощью которого можно определить уязвимость удалённых хостов. Основное назначение этой программы – определение состояние портов удалённого хоста (закрыты они, открыты или заблокированны). Также программа может на основании собственной базы знаний определить по поведению удалённого хоста, какая операционная система запущена на нём.

Команды удалённого терминала

Программа ssh является более современным и защищённым аналогом программы telnet.

Команды по управлению сетевым экраном

Команда **iptables** является интерфейсом к межсетевому экрану ядра Linux.

Контрольные вопросы

1. Какая команда используется для настройки сетевых интерфейсов?
2. Для чего используется команда route?

3. Расскажите о командах диагностики сети.
4. Какие существуют команды удалённого терминала?
5. Расскажите о команде iptables.