ИУ-10
Системное
Программное
Обеспечение
Администрирование Linux

Работа с сетью в Linux

Москва, 2022

На этом уроке

1. Обзор настроек, работающих на данный момент
2. Обзор утилит для получения информации о сети (ip, ifconfig)
3. Настройка сети в ОС Linux
4. Сбор сетевого дампа и его анализ

Содержание

Основы сети

IP адрес

Для настройки сети серверу нужен уникальный адрес и для этого используются IP-адреса. В настоящее время актуальны две версии:

- Адреса IPv4: основаны на 32-битных значениях и имеют четыре октета, разделенных точками, например *192.168.10.100*.
- Адреса IPv6: основаны на 128-битных значениях и записываются в восьми группах шестнадцатеричных чисел по 16 бит каждая, разделенных двоеточиями. Адрес IPv6 может выглядеть как: *fe80:badb:abe01:45bc:34ad:1313:6723:8798*.

Набор IP-адресов объединяют в сеть, в пределах которой компьютеры взаимодействуют почти напрямую. Для связи между сетями используется маршрутизатор. Маршрутизатор - это машина (часто для этой цели создано специальное оборудование), которая соединяет сети друг с другом.

Чтобы узнать, к какой сети принадлежит компьютер, для каждого IP-адреса используется маска подсети. Маска подсети определяет, какая часть сетевого адреса указывает на сеть, а какая - на узел (сервер, компьютер).

MAC-адрес

IP-адреса - это адреса, которые позволяют узлам связываться с любым другим узлом в Интернете. Однако это не единственные используемые адреса. Каждая сетевая карта также имеет 6-байтовый MAC-адрес, предназначенный для использования в локальной сети (вплоть до первого обнаруженного маршрутизатора); они не могут использоваться для связи между узлами, находящимися в разных сетях. MAC-адреса важны, потому что они помогают компьютерам найти конкретную сетевую карту, которой принадлежит IP-адрес.

Примером MAC-адреса является *00:0c:29:7d:9b:17*. Обратите внимание, что каждый MAC-адрес состоит из двух частей. Первая половина - это идентификатор поставщика, а вторая - уникальный идентификатор узла. Идентификаторы поставщиков зарегистрированы, и, используя зарегистрированные идентификаторы поставщиков, можно выделить уникальные MAC-адреса.

Порт

На серверах Вы обычно будете запускать службы, такие как Веб-сервер или TFTP-сервер. Для идентификации этих служб используются адреса портов. У каждой службы есть определенный адрес порта, например порт 80 для протокола HTTP или порт 22 для Secure Shell (SSH), и при сетевой связи отправитель и получатель используют адреса портов. Таким образом, существует адрес порта назначения, а также адрес порта источника, участвующие в сетевых коммуникациях.

Управление конфигурацией сети

Сетевые адреса можно назначить двумя способами:

- Фиксированные IP-адреса: полезно для серверов, которые всегда должны быть доступны по одному и тому же IP-адресу.

- Динамически назначаемые IP-адреса: полезно для устройств конечных пользователей и для экземпляров в облачной среде. Для динамического назначения IP-адресов обычно используется сервер протокола динамической конфигурации хоста (DHCP).

Просмотр состояния сетевого стека

Параметры интерфейсов

Прежде чем вы настраивать сеть, вы должны знать, как проверить текущую конфигурацию. Полезной информацией является:

- IP-адрес и маска подсети

- Маршрутизация

- Наличие портов и сервисов

Чтобы проверить настройки сети, необходимо использовать утилиту ip. Утилита ip - это современная утилита, с помощью которой можно отслеживать многие аспекты сети.

- ip addr для настройки и мониторинга сетевых адресов.

- ip route для настройки и отслеживания информации о маршрутизации.

- IP-ссылку для настройки и мониторинга состояния сетевого соединения.

Рассмотрим вывод команды `ip addr show`

```
root@server:~# ip addr show

1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group
default qlen 1000

    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00

    inet 127.0.0.1/8 scope host lo

       valid_lft forever preferred_lft forever

    inet6 ::1/128 scope host

       valid_lft forever preferred_lft forever

2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state
UP group default qlen 1000

       link/ether 08:00:27:05:9e:a6 brd ff:ff:ff:ff:ff:ff

       inet 192.168.1.84/24 brd 192.168.1.255 scope global dynamic enp0s3

          valid_lft 83818sec preferred_lft 83818sec

       inet6 fe80::a00:27ff:fe05:9ea6/64 scope link

          valid_lft forever preferred_lft forever
```

В результате этой команды можно ознакомиться со списком всех сетевых интерфейсов в Linux. Обычно существуют как минимум два интерфейса, но в определенных случаях интерфейсов может быть намного больше. Некоторые процессы используют протокол IP для внутренней связи. По этой причине вы всегда найдете интерфейс **loopback**, с адресом 127.0.0.1. Важная часть вывода команды относится к встроенной карте Ethernet. Команда показывает следующие элементы:

- Текущее состояние: наиболее важной частью этой строки является **UP**, которое показывает, что эта сетевая карта в настоящее время включена и доступна.

- Конфигурация MAC-адреса: это уникальный MAC-адрес, который устанавливается для каждой сетевой карты. Вы можете увидеть сам MAC-адрес (08:00:27:05:9e:a6), а также соответствующий широковещательный адрес.

- Конфигурация IPv4: в этой строке отображается текущий установленный IP-адрес, а также используемая маска подсети. Вы также можете увидеть широковещательный адрес, который используется для этой

конфигурации сети. Обратите внимание, что на некоторых интерфейсах вы можете найти несколько адресов IPv4. `dynamic` сигнализирует об использовании протокола DHCP.

- Конфигурация IPv6: в этой строке отображается текущий адрес IPv6 и его конфигурация. Даже если вы ничего не настроили, каждый интерфейс автоматически получает адрес IPv6, который можно использовать для связи в локальной сети.

Команда `ip link show` в сочетании с параметром -s, покажет текущую статистику о переданных и полученных пакетах, а также обзор ошибок, возникших во время передачи пакетов.

```
root@server:~# ip -s link show

1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode
DEFAULT group default qlen 1000

    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00

    RX: bytes packets errors dropped overrun mcast

    13674      1430     0      0       0       0

    TX: bytes packets errors dropped carrier collsns

    13674      1430     0      0       0       0

2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state
UP mode DEFAULT group default qlen 1000

    link/ether 08:00:27:05:9e:a6 brd ff:ff:ff:ff:ff:ff

    RX: bytes packets errors dropped overrun mcast

    103290692 127189    0      33      0       1386

    TX: bytes packets errors dropped carrier collsns

    823135     6204     0      0       0        0
```

Примечание! *Подобную информацию можно получить с помощью команды `ifconfig`.*

Один из важных аспектов сети - маршрутизация, которая требуется для связи между узлами в отличающихся сетях. В каждой сети есть маршрутизатор по умолчанию (также называемый шлюзом), и вы можете увидеть, какой маршрутизатор используется в качестве маршрутизатора по умолчанию, используя команду `ip route show`.

```
root@server:~# ip route show

default via 192.168.1.254 dev enp0s3 proto dhcp src 192.168.1.84 metric
100

192.168.1.0/24 dev enp0s3 proto kernel scope link src 192.168.1.84

192.168.1.254 dev enp0s3 proto dhcp scope link src 192.168.1.84 metric 100
```

Самая важная часть - это первая строка. Она показывает, что маршрут по умолчанию проходит через ("via") IP-адрес **192.168.1.254**, а также показывает, что для адресации этого IP-адреса необходимо использовать сетевой интерфейс **enp0s3**, и что этот маршрут был назначен DHCP-сервером. Метрика используется в случае, если к одному месту назначения доступно несколько маршрутов. Маршрут с наименьшей метрикойа предпочтителен.

Последующие строки идентифицируют локальные подключенные сети. При загрузке запись также добавляется для каждой локальной сети, и в этом примере они относятся к сетям **192.168.1.0/24**. Эти маршруты генерируются автоматически и не нуждаются в управлении.

Конфигурация сети

С помощью утилит можно не только просматривать настройки но и изменять их. К примеру, `ifconfig eth0 192.168.0.100 netmask 255.255.255.0` назначит новый IP адрес, а `route add default gw 192.168.0.254` выставит шлюз по умолчанию. Все настройки, выполненные таким образом, будут удалены в момент перезагрузки. Для "постоянной" конфигурации используется Netplan.

Netplan

Netplan - это утилита для простой настройки сети в системе Linux. Описание требуемых сетевых интерфейсов и того, как каждый из них должен работать, производится с помощью формата YAML. Yet Another Markup Language (YAML) - «дружественный» формат сериализации данных, основное внимание в котором необходимо обращать на отступы. Из этого описания Netplan генерирует всю необходимую конфигурацию для выбранного вами инструмента рендеринга.

Конфигурация создается администраторами, установщиками или другими способами развертывания ОС и находится в каталоге **/etc/netplan/**.

Во время загрузки Netplan считывает содержимое файлов и генерирует определенные конфигурацию серверной части в /**run**, чтобы передать управление устройствами сетевому демону. Давайте рассмотрим конфигурацию нашего сервера:

```
root@server:~# cat /etc/netplan/00-installer-config.yaml
# This is the network config written by 'subiquity'
network:
  ethernets:
    enp0s3:
      dhcp4: true
  version: 2
```

- network — это блок начало конфигурации
- renderer: networkd — сетевой менеджер, который будет использоваться для обработки (networkd или Network Manager)
- version: 2 — версия формата YAML
- ethernets: — блок говорит о том, что описываются ethernet интерфейсы.
- enp0s3: — имя сетевого адаптера.
- dhcp4: — состояние протокола DHCP

Как можно понять, на интерфейсах включен протокол DHCP. Использование этого протокола полезно для устройств конечных пользователей и для экземпляров в облачной среде, но для серверных версий рекомендуется использовать статическое назначение адресов.

Важно! *Перед началом изменения настроек сети рекомендуется сделать бэкап существующих файлов*.

Изменим содержимое существующего файла, после чего применим с помощью `netplan apply`. Ответ на ввод команды будет получен только в случае наличия ошибок в файле.

```
network:
    version: 2
    renderer: networkd
    ethernets:
        enp0s3:
            addresses:
                - 192.168.1.84/24
                - 192.168.1.87/24
            gateway4: 192.168.1.254
```

```
            nameservers:
                addresses:
                  - 192.168.1.254

root@server:~# netplan apply
root@server:~#
```

С помощью опций addresses и gateway4 указываются адреса сетевой карты и шлюза по умолчанию. В блоке nameservers отображены настройки DNS серверов. В ip addr show исчезла пометка dynamic, и теперь на интерфейсе назначены сразу два адреса.

```
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state
UP group default qlen 1000
    link/ether 08:00:27:05:9e:a6 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.84/24 brd 192.168.1.255 scope global enp0s3
        valid_lft forever preferred_lft forever
    inet 192.168.1.87/24 brd 192.168.1.255 scope global secondary enp0s3
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe05:9ea6/64 scope link
        valid_lft forever preferred_lft forever
```

С помощью netplan можно по необходимости настраивать и маршрутизацию.

```
root@server:~# cat /etc/netplan/00-installer-config.yaml
network:
    version: 2
    renderer: networkd
    ethernets:
        enp0s3:
            addresses:
                - 192.168.1.84/24
            gateway4: 192.168.1.254
            nameservers:
                addresses:
                  - 192.168.1.254
            routes:
                 - to: 8.8.8.8/32
                   via: 192.168.1.254
                   metric: 200

root@server:~# ip route show
default via 192.168.1.254 dev enp0s3 proto static
default via 192.168.1.254 dev enp0s3 proto static metric 200
192.168.1.0/24 dev enp0s3 proto kernel scope link src 192.168.1.84
```

После добавления маршрутов в выводе ip route show появилась вторая запись с метрикой 200.

Генерируемые имена не всегда несут смысловую нагрузку. Для их изменения можно также воспользоваться возможностями netplan. В файл

понадобится добавить критерии соотношений (match) конфигурации с реальным устройство через MAC-адрес и указание имя в параметр set-name .

```
root@server:~# cat /etc/netplan/00-installer-config.yaml
network:
    version: 2
    renderer: networkd
    ethernets:
        enp0s3:
            match:
                macaddress: 08:00:27:05:9e:a6
            set-name: eth0
            addresses:
                - 192.168.1.84/24
            gateway4: 192.168.1.254
            nameservers:
                addresses:
                  - 192.168.1.254
            routes:
                 - to: 8.8.8.8/0
                   via:
                   192.168.1.254
                   metric: 200
root@server:~# ip a show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group
default qlen 1000
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state
UP group default qlen 1000
    link/ether 08:00:27:05:9e:a6 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.84/24 brd 192.168.1.255 scope global eth0
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe05:9ea6/64 scope link
        valid_lft forever preferred_lft forever
```

Примечание! *Для изменения имени понадобится перезагрузка системы.*

Проверка работы сети

Ping

Ping - это служебная программа, используемая для проверки доступности хоста в сети по IP адресу. Эта утилита доступна практически для всех операционных систем, имеющих сетевые возможности, в том числе и Linux. Ping измеряет RTT(Round-Trip Time) сообщений, отправленных с исходного компьютера на конечный хост.

```
root@server:\sim# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=109 time=22.6 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=109 time=26.1 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=109 time=23.9 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=109 time=24.2 ms
^C
--- 8.8.8.8 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3008ms
rtt min/avg/max/mdev = 22.641/24.218/26.094/1.236 ms
```

По умолчанию утилита будет отправлять сообщения, пока не будет передана команда на прерывание. С помощью ключа **-c** можно указать ограничение, а с помощью **-i** интервал.

```
root@server:~# ping -i 0.1 -c 5 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=109 time=23.8 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=109 time=23.0 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=109 time=23.8 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=109 time=23.6 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=109 time=23.5 ms

--- 8.8.8.8 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 404ms
rtt min/avg/max/mdev = 23.025/23.528/23.780/0.272 ms
```

Порты приложений

Сетевые проблемы могут быть связаны с локальным IP-адресом и настройками маршрутизатора, но также могут быть связаны с сетевыми портами, которые недоступны на вашем сервере или на удаленном сервере. Чтобы проверить доступность портов на вашем сервере, вы можете использовать команду `netstat` или более новую команду `ss`, которая обеспечивает ту же функциональность.

```
root@server:~# netstat -tulpan
Active Internet connections (servers and
Proto Recv-Q Send-Q Local Address           Foreign                 Stat
PID/Program name
tcp          0      0 127.0.0.53:53          0.0.0.0:*
             LISTEN
582/systemd-resolve
tcp          0      0 0.0.0.0:22             0.0.0.0:*
             LISTEN
633/sshd: /usr/sbin
tcp          0      0 192.168.1.84:22      192.168.1.73:60111
633/sshd: /usr/sbin
udp          0                 0 127.0.0.53:53                    0.0.0.0:
582/systemd-resolve
udp6                   0                 0 fe80::a00:27ff:fe05:546    :::*
580/systemd-network
root@server:~#
```

Набрав `ss -lt`, вы увидите все прослушивающие TCP-порты в локальной системе

```
root@server:~# ss -lt
State               Recv-Q               Send-Q                     Local
Address:Port                            Peer                     Process
LISTEN                           0                                   4096
127.0.0.53%lo:domain                                  0.0.0.0:
LISTEN                           0                                    128
0.0.0.0:ssh                                      0.0.0.0:
LISTEN                           0                                    128
[::]:ssh                                         [::]:
```

Обратите внимание, какие из портов прослушиваются. Некоторые порты прослушивают только адрес loopback IPv4 127.0.0.1 или IPv6 :: 1, что означает, что они доступны только локально и недоступны с внешних компьютеров. Другие порты прослушивают *, что означает все адреса IPv4, или ::: *, который представляет все порты на всех адресах IPv6.

Netcat

Для проверки портов на удаленном сервере следует воспользоваться утилитой Netcat и командой nc. Netcat может осуществлять соединение по TCP портам, а также посылать и принимать TCP,UDP трафик.

Для простой проверки доступности удаленного порта воспользуйтесь `nc -zvw1`.

```
root@server:~# nc -zvw1 8.8.8.8 80
nc: connect to 8.8.8.8 port 80 (tcp) timed out: Operation now in progress
root@server:~# nc -zvw1 8.8.8.8 53
Connection to 8.8.8.8 53 port [tcp/domain] succeeded!
```

Чтобы послать послать UDP пакеты, воспользуйтесь опцией -u.

```
root@server:~# root@server:~# nc -u 8.8.8.8 53
Sending some packet to GOOGLE DNS!
^C
```

Tcpdump

Tcpdump является самой мощной утилитой для анализа работы сетевого стека. Эта утилита выводит описание содержимого пакетов сетевого интерфейса.

```
root@server:~# tcpdump
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
13:43:42.366813   IP   server.ssh   >   192.168.1.73.60111:   Flags
                  seq
3538046563:3538046671,  ack 2391002831,  win 501,  options [nop,nop,TS val
1982379078 ecr 802292551], length 108
13:43:42.366891 IP server.ssh > 192.168.1.73.60111: Flags [P.], seq
108:244, ack 1, win 501, options [nop,nop,TS val 1982379078 ecr
802292551], length 136
13:43:42.367002 IP 192.168.1.73.60111 > server.ssh: Flags [.], ack 108,
win 2046, options [nop,nop,TS val 802292585 ecr 1982379078], length 0
13:43:42.367002 IP 192.168.1.73.60111 > server.ssh: Flags [.], ack 244,
win 2044, options [nop,nop,TS val 802292585 ecr 1982379078], length 0
13:43:42.367059 IP server.ssh > 192.168.1.73.60111: Flags [P.], seq
244:272, ack 1, win 501, options [nop,nop,TS val 1982379079 ecr
802292585], length 28
```

Описанию параметров пакета предшествует отметка времени, которая по умолчанию печатается в виде часов, минут, секунд и долей секунды после полуночи.Работой tcpdump можно управлять с помощью набора ключей и параметров.

В случае наличия нескольких интерфейсов с помощью ключа -i можно выбрать один, а с параметром `host` определить адрес, трафик до которого необходимо отобразить на экране.

```
root@server:~# tcpdump -i eth0 host 8.8.8.8
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
13:47:44.555289 IP server > dns.google: ICMP echo request, id 4, seq 356,
length 64
```

```
13:47:44.578818 IP dns.google > server: ICMP echo reply, id 4, seq 356,
length 64
13:47:45.558374 IP server > dns.google: ICMP echo request, id 4, seq 357,
length 64
13:47:45.583456 IP dns.google > server: ICMP echo reply, id 4, seq 357,
length 64
13:47:46.559604 IP server > dns.google: ICMP echo request, id 4, seq 358,
length 64
13:47:46.581900 IP dns.google > server: ICMP echo reply, id 4, seq 358,
length 64
13:47:47.561489 IP server > dns.google: ICMP echo request, id 4, seq 359,
length 64
13:47:47.584801 IP dns.google > server: ICMP echo reply, id 4, seq 359,
length 64

Src dst
```

Фильтрации по адресу источника и назначения производится с помощью параметров `Src` и `dst` соответственно. В пример ниже видны только ответы от **8.8.8.8** , но не видно запросов.

```
root@server:~# tcpdump -i eth0 src 8.8.8.8
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
13:51:58.094222  IP dns.google > server: ICMP echo reply, id 4, seq 609,
64
13:51:59.099626  IP dns.googl  > server  ICMP echo reply, id 4, seq 610,
64
13:52:00.100698  IP dns.googl  > server  ICMP echo reply, id 4, seq 611,
64
```

Описанные действия крайне удобны для сбора трафика, но в tcpdump существует возможность комбинировать варианты, чтобы выделить именно те пакеты, которые требуются. Есть три способа создания комбинаций, схожих с базовым программированием:

- И - **and** или **&&**

- ИЛИ - **or** или **||**

- Исключить - **not** или **!**

С помощью `and` объединили условия адреса и порта.

```
root@server:~# tcpdump -i eth0 host 8.8.8.8 and port 53
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
13:54:09.475970 IP server.38878 > dns.google.domain: Flags [S], seq
1882266456, win 64240, options [mss 1460,sackOK,TS val 2494499897 ecr 0,
nop,wscale 7], length 0
13:54:09.500842 IP dns.google.domain > server.38878: Flags [S.], seq
1845975136, ack 1882266457, win 65535, options [mss 1430,sackOK,TS val
```

```
341650861 ecr 2494499897,nop,wscale 8], length 0
13:54:09.500895 IP server.38878 > dns.google.domain: Flags [.], ack 1, win
502, options [nop,nop,TS val 2494499921 ecr 341650861], length 0
13:54:11.523621 IP dns.google.domain > server.38878: Flags [F.], seq 1,
ack 1, win 256, options [nop,nop,TS val 341652884 ecr 2494499921], length
0
13:54:11.523843 IP server.38878 > dns.google.domain: Flags [F.], seq 1,
ack 2, win 502, options [nop,nop,TS val 2494501944 ecr 341652884], length
0
13:54:11.547083 IP dns.google.domain > server.38878: Flags [.], ack 2, win
256, options [nop,nop,TS val 341652908 ecr 2494501944], length 0
^C
6 packets captured
6 packets received by filter
0 packets dropped by kernel
```

С помощью исключения порта 22 собирается весь трафик, но без сессий SSH, которые часто только мешают в анализе пакетов.

```
root@server:~# tcpdump -i eth0 port ! 22
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
14:12:11.118891 IP 192.168.1.75.50222 > 224.0.0.252.hostmon: UDP, length
22
14:12:11.119213   IP   server.42847   >   mygpon.domain:   25616+
                       [1au] PTR? 255.1.168.192.in-addr.arpa. (55)
14:12:11.126544 IP mygpon.domain > server.42847: 25616 NXDomain 0/1/1(104)
14:12:11.127434    IP     server.42847    >     mygpon.domain:    25616+
PTR? 255.1.168.192.in-addr.arpa. (44)
14:12:11.130657 IP mygpon.domain > server.42847: 25616 NXDomain 0/0/0 (44)
14:12:11.131321   IP   server.41849   >   mygpon.domain:   13992+   [1au]
                  PTR? 75.1.168.192.in-addr.arpa. (54)
14:12:11.138647 IP mygpon.domain > server.41849: 13992 NXDomain 0/1/1(103)
14:12:11.139065    IP     server.41849    >     mygpon.domain:    13992+
                   PTR? 75.1.168.192.in-addr.arpa. (43)
^C
6 packets captured
6 packets received by filter
0 packets dropped by kernel
```

Важно! *С помощью опции -w <имя файла> трафик будет записан в указанный файл, а ключ -s0 уберет ограничение на размер записываемых пакетов.*

Практическое задание

1. С помощью утилиты netplan повторить имеющиеся в системе параметры, но задать их статически.

2. Запустить ping до сайта ya.ru, проверить доступность.

3. Не останавливая ping, открыть отдельное окно терминала и вывести на экран только запросы в сторону (echo request) в сторону сайта ya.ru.

4. С помощью утилиты nc проверить доступность основных портов (53,80, 443) у сайта ya.ru, habr.com, google.com .

5. Проанализировать предложенный .pcap файл дампа сетевого трафика.

6. Установить утилиту ssh, подключиться с хостовой машины к виртуальной машине по ssh (прим. необходимо сменить конфигурацию сетевого адаптера гостевой ОС). Отключить возможность аутентифицироваться пользователю root. Сделать аутентификацию с использованием публичного ключа.

Глоссарий

IP-адрес — уникальный сетевой адрес узла в компьютерной сети, построенной на основе стека протоколов TCP/IP.

Порт - сетевой идентификатор приложения.

MAC адрес - адрес сетевой карты.

Netplan - утилита для настройки сети в Linux.

Дополнительные материалы

Статья о netplan

Используемые источники

https://ru.wikipedia.org/wiki/Динамический_порт
https://netplan.io/examples/