Отчёт по лабораторной работе «Локальные сети»

Овчинников Владислав Александрович

11 декабря 2020 г.

Содержание

1. Получение адреса по DHCP

Для динамической настройки IP-адреса и всех его параметров может использоваться протокол DHCP. Он позволяет выдавать компьютерам в одном сегменте широковещания IP-адреса из некоторого указанного диапазона. Кроме адреса, клиент DHCP получает от сервера маску сети, адрес (или адреса) маршрутизаторов, адреса DNS-серверов. Протокол DHCP использует широковещательную рассылку поверх протокола UDP для обмена данными между клиентом и сервером. Поскольку у клиента еще нет IP-адреса, в качестве него используется 0.0.0.0, а IP-адресом получателя при широковещательном обмене выступает 255.255.255.255.

Для изучения работы протокола DHCP в случае выдачи случайного IP-адреса был запущен DHCP-сервер на маршрутизаторе **r2** с помощью команды **service dhcp3-server start**. Данный сервер настроен так, чтобы клиенты получали динамические IP-адреса из диапазона, заданного в конфигурационном файле **/etc/dhcp3/dhcpd.conf**.

Конфигурационный файл DHCP-сервера на маршрутизаторе **r2**:

```
subnet 172.16.0.0 netmask 255.255.0.0 {}

subnet 10.20.0.0 netmask 255.255.0.0
{
  range 10.20.0.2 10.20.0.200;
```

```
  option routers 10.20.0.1;
  option domain-name-servers 10.20.0.1;
}
```

Дамп собирался при помощи программы **tcpdump** на маршрутизаторе **r2**, запущенной с параметрами **-tenv -s 1000**.

```
10:10:10:10:10:ee > ff:ff:ff:ff:ff:ff, ethertype IPv4 (0x0800), length 342:
        (tos 0x10, ttl 128, id 0, offset 0, flags [none], proto UDP (17), length 328)
        0.0.0.0.68 > 255.255.255.255.67: BOOTP/DHCP, Request from 10:10:10:10:10:ee,
        length 300, xid 0x7b8f9a6c, Flags [none]
          Client-Ethernet-Address 10:10:10:10:10:ee
          Vendor-rfc1048 Extensions
            Magic Cookie 0x63825363
            DHCP-Message Option 53, length 1: Discover
            Parameter-Request Option 55, length 12:
              Subnet-Mask, BR, Time-Zone, Default-Gateway
              Domain-Name, Domain-Name-Server, Option 119, Hostname
              Netbios-Name-Server, Netbios-Scope, MTU, Classless-Static-Route
3a:40:ee:31:9e:cd > 10:10:10:10:10:ee, ethertype IPv4 (0x0800), length 342:
        (tos 0x10, ttl 128, id 0, offset 0, flags [none], proto UDP (17), length 328)
        10.20.0.1.67 > 10.20.0.2.68: BOOTP/DHCP, Reply,
        length 300, xid 0x7b8f9a6c, Flags [none]
          Your-IP 10.20.0.2
          Client-Ethernet-Address 10:10:10:10:10:ee
          Vendor-rfc1048 Extensions
            Magic Cookie 0x63825363
            DHCP-Message Option 53, length 1: Offer
            Server-ID Option 54, length 4: 10.20.0.1
            Lease-Time Option 51, length 4: 43200
            Subnet-Mask Option 1, length 4: 255.255.0.0
            Default-Gateway Option 3, length 4: 10.20.0.1
            Domain-Name-Server Option 6, length 4: 10.20.0.1
10:10:10:10:10:ee > ff:ff:ff:ff:ff:ff, ethertype IPv4 (0x0800), length 342:
        (tos 0x10, ttl 128, id 0, offset 0, flags [none], proto UDP (17), length 328)
        0.0.0.0.68 > 255.255.255.255.67: BOOTP/DHCP, Request from 10:10:10:10:10:ee,
        length 300, xid 0x7b8f9a6c, Flags [none]
          Client-Ethernet-Address 10:10:10:10:10:ee
          Vendor-rfc1048 Extensions
            Magic Cookie 0x63825363
            DHCP-Message Option 53, length 1: Request
            Server-ID Option 54, length 4: 10.20.0.1
            Requested-IP Option 50, length 4: 10.20.0.2
            Parameter-Request Option 55, length 12:
              Subnet-Mask, BR, Time-Zone, Default-Gateway
              Domain-Name, Domain-Name-Server, Option 119, Hostname
              Netbios-Name-Server, Netbios-Scope, MTU, Classless-Static-Route
3a:40:ee:31:9e:cd > 10:10:10:10:10:ee, ethertype IPv4 (0x0800), length 342:
        (tos 0x10, ttl 128, id 0, offset 0, flags [none], proto UDP (17), length 328)
        10.20.0.1.67 > 10.20.0.2.68: BOOTP/DHCP, Reply,
        length 300, xid 0x7b8f9a6c, Flags [none]
```

```
          Your-IP 10.20.0.2
          Client-Ethernet-Address 10:10:10:10:10:ee
          Vendor-rfc1048 Extensions
            Magic Cookie 0x63825363
            DHCP-Message Option 53, length 1: ACK
            Server-ID Option 54, length 4: 10.20.0.1
            Lease-Time Option 51, length 4: 43200
            Subnet-Mask Option 1, length 4: 255.255.0.0
            Default-Gateway Option 3, length 4: 10.20.0.1
            Domain-Name-Server Option 6, length 4: 10.20.0.1
```

В начале видно, что клиент посылает широковещательный запрос на обнаружение доступных DHCP-серверов (на это указывает MAC-адрес, IP-адрес и порт получателя сообщения, также на это указывает тип сообщения Discover). Затем сервер отправляет сообщение клиенту с предложением конфигурации (на это опять указывают адреса получателя, а также тип сообщения Offer). Выбрав одну из конфигураций клиент отправляет запрос (тип сообщения Request). Он рассылается широковещательно, уведомляя всех в сети какая конфигурация, кем и где занята. В завершении сервер подтверждает запрос и отсылает клиенту сообщение с подтверждением (тип сообщения ACK).

Вывод на экран запросов и ответов DHCP-клиента на машине **ws21**.

```
Listening on LPF/eth0/10:10:10:10:10:ee
Sending on   LPF/eth0/10:10:10:10:10:ee
Sending on   Socket/fallback
DHCPDISCOVER on eth0 to 255.255.255.255 port 67 interval 8
DHCPOFFER from 10.20.0.1
DHCPREQUEST on eth0 to 255.255.255.255 port 67
DHCPACK from 10.20.0.1
bound to 10.20.0.2 -- renewal in 17504 seconds.
done.
```

Для изучения работы протокола DHCP был запущен DHCP-сервер на маршрутизаторе **r1**. Данный сервер настроен так, чтобы клиенты получали статические IP-адреса, заданные в конфигурационном файле.

Конфигурационный файл данного DHCP-сервера, запущенного на маршрутизаторе **r1**:

```
subnet 172.16.0.0 netmask 255.255.0.0 {}

subnet 10.10.0.0 netmask 255.255.0.0
{
  range 10.10.0.2 10.10.10.200;
  option routers 10.10.0.1;
  option domain-name-servers 10.10.0.1;
}

host ws11 {
    hardware ethernet 10:10:10:10:10:BA;
    fixed-address 10.10.1.1;
}
```

```
host ws12 {
    hardware ethernet 10:10:10:10:10:BB;
    fixed-address 10.10.2.1;
}

host ws13 {
    hardware ethernet 10:10:10:10:10:BC;
    fixed-address 10.10.3.1;
}


host ws14 {
    hardware ethernet 10:10:10:10:10:BD;
    fixed-address 10.10.4.1;
}
host s11 {
    hardware ethernet 10:10:10:10:20:AA;
    fixed-address 10.10.4.10;
}

host s12 {
    hardware ethernet 10:10:10:10:20:BB;
    fixed-address 10.10.4.20;
}

host s13 {
    hardware ethernet 10:10:10:10:20:CC;
    fixed-address 10.10.4.30;
}
```

Дамп собирался при помощи программы **tcpdump** на маршрутизаторе **r1**, запущенной с параметрами **-tenv -s 1000**.

Получение фиксированного адреса с помощью DHCP:

```
10:10:10:10:20:cc > ff:ff:ff:ff:ff:ff, ethertype IPv4 (0x0800), length 342:
        (tos 0x10, ttl 128, id 0, offset 0, flags [none], proto UDP (17), length 328)
        0.0.0.0.68 > 255.255.255.255.67: BOOTP/DHCP, Request from 10:10:10:10:20:cc,
        length 300, xid 0xb391540a, Flags [none]
          Client-Ethernet-Address 10:10:10:10:20:cc
          Vendor-rfc1048 Extensions
            Magic Cookie 0x63825363
            DHCP-Message Option 53, length 1: Discover
            Requested-IP Option 50, length 4: 10.10.4.30
            Parameter-Request Option 55, length 12:
              Subnet-Mask, BR, Time-Zone, Default-Gateway
              Domain-Name, Domain-Name-Server, Option 119, Hostname
              Netbios-Name-Server, Netbios-Scope, MTU, Classless-Static-Route
0e:ab:f8:0c:10:4b > 10:10:10:10:20:cc, ethertype IPv4 (0x0800), length 342:
        (tos 0x10, ttl 128, id 0, offset 0, flags [none], proto UDP (17), length 328)
        10.10.0.1.67 > 10.10.4.30.68: BOOTP/DHCP, Reply,
        length 300, xid 0xb391540a, Flags [none]
```

```
          Your-IP 10.10.4.30
          Client-Ethernet-Address 10:10:10:10:20:cc
          Vendor-rfc1048 Extensions
            Magic Cookie 0x63825363
            DHCP-Message Option 53, length 1: Offer
            Server-ID Option 54, length 4: 10.10.0.1
            Lease-Time Option 51, length 4: 43200
            Subnet-Mask Option 1, length 4: 255.255.0.0
            Default-Gateway Option 3, length 4: 10.10.0.1
            Domain-Name-Server Option 6, length 4: 10.10.0.1
10:10:10:10:20:cc > ff:ff:ff:ff:ff:ff, ethertype IPv4 (0x0800), length 342:
        (tos 0x10, ttl 128, id 0, offset 0, flags [none], proto UDP (17), length 328)
        0.0.0.0.68 > 255.255.255.255.67: BOOTP/DHCP, Request from 10:10:10:10:20:cc,
        length 300, xid 0xb391540a, Flags [none]
          Client-Ethernet-Address 10:10:10:10:20:cc
          Vendor-rfc1048 Extensions
            Magic Cookie 0x63825363
            DHCP-Message Option 53, length 1: Request
            Server-ID Option 54, length 4: 10.10.0.1
            Requested-IP Option 50, length 4: 10.10.4.30
            Parameter-Request Option 55, length 12:
              Subnet-Mask, BR, Time-Zone, Default-Gateway
              Domain-Name, Domain-Name-Server, Option 119, Hostname
              Netbios-Name-Server, Netbios-Scope, MTU, Classless-Static-Route
0e:ab:f8:0c:10:4b > 10:10:10:10:20:cc, ethertype IPv4 (0x0800), length 342:
        (tos 0x10, ttl 128, id 0, offset 0, flags [none], proto UDP (17), length 328)
        10.10.0.1.67 > 10.10.4.30.68: BOOTP/DHCP, Reply,
        length 300, xid 0xb391540a, Flags [none]
          Your-IP 10.10.4.30
          Client-Ethernet-Address 10:10:10:10:20:cc
          Vendor-rfc1048 Extensions
            Magic Cookie 0x63825363
            DHCP-Message Option 53, length 1: ACK
            Server-ID Option 54, length 4: 10.10.0.1
            Lease-Time Option 51, length 4: 43200
            Subnet-Mask Option 1, length 4: 255.255.0.0
            Default-Gateway Option 3, length 4: 10.10.0.1
            Domain-Name-Server Option 6, length 4: 10.10.0.1
```

Вывод на экран запросов и клиентов DHCP-клиента на машине **s11**:

```
Listening on LPF/eth0/10:10:10:10:20:aa
Sending on   LPF/eth0/10:10:10:10:20:aa
Sending on   Socket/fallback
DHCPDISCOVER on eth0 to 255.255.255.255 port 67 interval 3
DHCPOFFER from 10.10.0.1
DHCPREQUEST on eth0 to 255.255.255.255 port 67
DHCPACK from 10.10.0.1
bound to 10.10.4.10 -- renewal in 16950 seconds.
done.
```

Получение статического или динамического IP-адреса с точки зрения запросов ничем не отличаются. Разница состоит лишь в том, что IP-адреса в одном случае привязаны к MAC-адресу клиента, а в другом нет.

Когда DHCP-клиент пытается обнаружить в течение некоторого времени DHCP-сервер в сети, но не находит его, он засыпает:

```
Listening on LPF/eth0/10:10:10:10:20:aa
Sending on   LPF/eth0/10:10:10:10:20:aa
Sending on   Socket/fallback
DHCPDISCOVER on eth0 to 255.255.255.255 port 67 interval 8
DHCPDISCOVER on eth0 to 255.255.255.255 port 67 interval 7
DHCPDISCOVER on eth0 to 255.255.255.255 port 67 interval 10
DHCPDISCOVER on eth0 to 255.255.255.255 port 67 interval 9
DHCPDISCOVER on eth0 to 255.255.255.255 port 67 interval 9
DHCPDISCOVER on eth0 to 255.255.255.255 port 67 interval 11
DHCPDISCOVER on eth0 to 255.255.255.255 port 67 interval 7
No DHCPOFFERS received.
No working leases in persistent database - sleeping.
done.
```

2. Использование VPN

VPN - это обобщенное название технологий, позволяющих обеспечить одно или несколько сетевых соединений (логическую сеть) поверх другой сети (например, Интернет).

В данной лабораторной работе используется OpenVPN. OpenVPN - это свободная реализация технологии виртуальной частной сети (VPN) с открытым исходным кодом для создания зашифрованных каналов типа точка-точка или сервер-клиенты между компьютерами. Она позволяет устанавливать соединения между компьютерами, находящимися за NAT и сетевым экраном, без необходимости изменения их настроек.

На маршрутизаторах **r1** и **r2** был настроен OpenVPN с помощью конфигурационного файла и в последствии запущен с помощью команды **service openvpn start**.

Содержимое конфигурационного файла **/etc/openvpn/tun0.conf** на маршрутизаторе **r2**:

```
remote 172.16.1.3 1194
proto udp
dev tun

ifconfig 10.100.100.2 10.100.100.1
secret /etc/openvpn/keys/somesecret.key

status /var/log/openvpn/tun0.status
log /var/log/openvpn/tun0.log
```

Содержимое конфигурационного файла **/etc/openvpn/tun0.conf** на маршрутизаторе **r1**:

```
local 172.16.1.3
proto udp
```

```
port 1194
dev tun

ifconfig 10.100.100.1 10.100.100.2
secret /etc/openvpn/keys/somesecret.key

status /var/log/openvpn/tun0.status
log /var/log/openvpn/tun0.log
```

Протокол RIP используется для получения маршрутной информации для всей виртуальной сети. Сообщения RIP передаются также через виртуальную частную сеть, организованную при помощи OpenVPN.

Таблица маршрутизации **r1** после запуска OpenVPN и работы RIP:

```
10.100.100.2 dev tun0  proto kernel  scope link  src 10.100.100.1
10.20.0.0/16 via 10.100.100.2 dev tun0  proto zebra  metric 2
10.10.0.0/16 dev eth0  proto kernel  scope link  src 10.10.0.1
172.16.0.0/16 dev eth1  proto kernel  scope link  src 172.16.1.3
default via 172.16.1.2 dev eth1
```

Таблица маршрутизации **r2** после запуска OpenVPN и работы RIP:

```
10.100.100.1 dev tun0  proto kernel  scope link  src 10.100.100.2
10.20.0.0/16 dev eth0  proto kernel  scope link  src 10.20.0.1
10.10.0.0/16 via 10.100.100.1 dev tun0  proto zebra  metric 2
172.16.0.0/16 dev eth1  proto kernel  scope link  src 172.16.1.4
default via 172.16.1.2 dev eth1
```

Назначенные IP-адреса на интерфейсах маршрутизатора **r1**:

```
1: lo: <LOOPBACK,UP,LOWERUP> mtu 16436 qdisc noqueue
    inet 127.0.0.1/8 scope host lo
3: eth1: <BROADCAST,MULTICAST,UP,LOWERUP> mtu 1500 qdisc pfifofast qlen 1000
    inet 172.16.1.3/16 brd 172.16.255.255 scope global eth1
4: eth0: <BROADCAST,MULTICAST,UP,LOWERUP> mtu 1500 qdisc pfifofast qlen 1000
    inet 10.10.0.1/16 brd 10.10.255.255 scope global eth0
5: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWERUP> mtu 1500 qdisc pfifofast qlen 100
    inet 10.100.100.1 peer 10.100.100.2/32 scope global tun0
```

Назначенные IP-адреса на интерфейсах маршрутизатора **r2**:

```
1: lo: <LOOPBACK,UP,LOWERUP> mtu 16436 qdisc noqueue
    inet 127.0.0.1/8 scope host lo
3: eth1: <BROADCAST,MULTICAST,UP,LOWERUP> mtu 1500 qdisc pfifofast qlen 1000
    inet 172.16.1.4/16 brd 172.16.255.255 scope global eth1
4: eth0: <BROADCAST,MULTICAST,UP,LOWERUP> mtu 1500 qdisc pfifofast qlen 1000
    inet 10.20.0.1/16 brd 10.20.255.255 scope global eth0
5: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWERUP> mtu 1500 qdisc pfifofast qlen 100
    inet 10.100.100.2 peer 10.100.100.1/32 scope global tun0
```

Дамп RIP-сообщений на маршрутизаторе **r1** собирался с помощью команды **tcpdump -tnv -i tun0 -s 1528 udp**:

```
IP (tos 0x0, ttl 1, id 0, offset 0, flags [DF], proto UDP (17), length 52)
        10.100.100.1.520 > 224.0.0.9.520:
        RIPv2, Response, length: 24, routes: 1
          AFI: IPv4:       10.10.0.0/16, tag 0x0000, metric: 1, next-hop: self
IP (tos 0x0, ttl 1, id 0, offset 0, flags [DF], proto UDP (17), length 52)
        10.100.100.1.520 > 224.0.0.9.520:
        RIPv2, Response, length: 24, routes: 1
          AFI: IPv4:       10.10.0.0/16, tag 0x0000, metric: 1, next-hop: self
IP (tos 0x0, ttl 1, id 0, offset 0, flags [DF], proto UDP (17), length 52)
        10.100.100.2.520 > 224.0.0.9.520:
        RIPv2, Response, length: 24, routes: 1
          AFI: IPv4:       10.20.0.0/16, tag 0x0000, metric: 1, next-hop: self
IP (tos 0x0, ttl 1, id 0, offset 0, flags [DF], proto UDP (17), length 52)
        10.100.100.1.520 > 224.0.0.9.520:
        RIPv2, Response, length: 24, routes: 1
          AFI: IPv4:       10.10.0.0/16, tag 0x0000, metric: 1, next-hop: self
```

Была проверена работа VPN с помощью проверки маршрута от **ws21**, находящегося в сети **10.20.0.0/16**, к которой подключен маршрутизатор **r2**, до **s11**, находящегося в сети **10.10.0.0/16**, к которой подключен маршрутизатор **r1**.

```
traceroute to 10.10.4.10 (10.10.4.10), 64 hops max, 40 byte packets
 1  10.20.0.1 (10.20.0.1)  1 ms  1 ms  1 ms
 2  10.100.100.1 (10.100.100.1)  2 ms  1 ms  1 ms
 3  10.10.4.10 (10.10.4.10)  3 ms  2 ms  1 ms
```

Также был собран дамп ICMP-сообщений с помощью **tcpdump** на интерфейсе **tun0**:

```
tcpdump: listening on tun0, link-type RAW (Raw IP), capture size 96 bytes
IP (tos 0x0, ttl 64, id 0, offset 0, flags [DF], proto ICMP (1), length 84)
        10.100.100.2 > 10.100.100.1: ICMP echo request, id 49410, seq 1, length 64
IP (tos 0x0, ttl 64, id 43803, offset 0, flags [none], proto ICMP (1), length 84)
        10.100.100.1 > 10.100.100.2: ICMP echo reply, id 49410, seq 1, length 64
IP (tos 0x0, ttl 64, id 0, offset 0, flags [DF], proto ICMP (1), length 84)
        10.100.100.2 > 10.100.100.1: ICMP echo request, id 49410, seq 2, length 64
IP (tos 0x0, ttl 64, id 43804, offset 0, flags [none], proto ICMP (1), length 84)
        10.100.100.1 > 10.100.100.2: ICMP echo reply, id 49410, seq 2, length 64
IP (tos 0x0, ttl 64, id 0, offset 0, flags [DF], proto ICMP (1), length 84)
        10.100.100.2 > 10.100.100.1: ICMP echo request, id 49410, seq 3, length 64
IP (tos 0x0, ttl 64, id 43805, offset 0, flags [none], proto ICMP (1), length 84)
        10.100.100.1 > 10.100.100.2: ICMP echo reply, id 49410, seq 3, length 64
```

Как можно видеть из логов, маршрут проходит через сеть **10.100.100.0/24**, что означает работу VPN.

3. Правила фильтации пакетов и трансляции адресов

Сетевой экран является частью компьютерного оборудования с аппаратным или программным обеспечением, который фильтрует входящие и исходящие сетевые пакеты (исходящие из/в локальную сеть) и при совпадении некоторых условий выполняет действия.

Ядро Linux содержит файрвол Netfilter. Управлять им можно из пространства пользователя с помощью команд iptables и ip6tables. Разница между этими двумя командами состоит в том, что первая работает с IPv4, тогда как последняя предназначена для IPv6.

Netfilter использует четыре различных таблицы, которые хранят правила, регулирующие три вида операций над пакетами: filter, NAT, mangle, raw. Каждая таблица содержит списки правил под назнванием цепочки (chains). Таблица filter имеет три стандартных цепи: INPUT, OUTPUT, FORWARD. Таблица nat также имеет три стандартных цепи: PREROUTING, POSTROUTING, OUTPUT. Также есть понятие действий при выполнении данных цепочек: ACCEPT, REJECT, DROP, LOG, ULOG, RETURN, SNAT, DNAT, MASQUERADE, REDIRECT, можно указать также "имя цепочки"для того, чтобы перейти к ней после выполнения текущей.

Был проведен эксперимент по запрету icmp-сообщений.

Сценарий фильтрации на маршрутизаторе **r1** (файл **/etc/firewall**):

```
#!/bin/sh
LAN=eth0
INET=eth1
VPN=tun0

# Удаление всех правил в таблице "filter" (по-умолчанию).
iptables -F

# Удаление правил в таблице "nat" (её надо указать явно).
iptables -F -t nat

# По-умолчанию все маршрутизируемые пакеты выбрасываются.
iptables --policy FORWARD DROP

# ICMP разрешим
# iptables -A FORWARD -p icmp -j ACCEPT

# Запрещаем icmp
iptables -A FORWARD -i $VPN -p icmp -j DROP
iptables -A FORWARD -o $VPN -p icmp -j DROP

# Разрешаем любую маршрутизацию для интерфейса VPN
iptables -A FORWARD -i $VPN -j ACCEPT
iptables -A FORWARD -o $VPN -j ACCEPT

# Включение SNAT для маршрутизируемых пакетов, выходящих
# через eth1. Это правило выполняется после самой маршрутизации
# (POSTROUTING) и помещается в таблицу правил "nat".
iptables -t nat -A POSTROUTING -o $INET -j MASQUERADE

# Разрешение пакетов-ответов (они отслеживаются как --state ESTABLISHED)
iptables -A FORWARD -m state --state ESTABLISHED -i $INET -j ACCEPT
```

Были выведены цепочки из таблицы **filter** с помощью команды **iptables -L -nv**:

```
Chain INPUT (policy ACCEPT 2319 packets, 192K bytes)
 pkts bytes target     prot opt in     out     source    destination
```

```
Chain FORWARD (policy DROP 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source    destination
    5   420 DROP       icmp --  tun0   *       0.0.0.0/0 0.0.0.0/0
    0     0 DROP       icmp --  *      tun0    0.0.0.0/0 0.0.0.0/0
   11   570 ACCEPT     all  --  tun0   *       0.0.0.0/0 0.0.0.0/0
    9   548 ACCEPT     all  --  *      tun0    0.0.0.0/0 0.0.0.0/0
    0     0 ACCEPT     all  --  eth1   *       0.0.0.0/0 0.0.0.0/0  state ESTABLISHED

Chain OUTPUT (policy ACCEPT 1681 packets, 139K bytes)
 pkts bytes target     prot opt in     out     source    destination
```

В данной таблице видно, что было применено 5 правил цепочки FORWARD, при этом правил цепочке INPUT и OUTPUT нет.

Были выведены цепочки из таблице **nat** с помощью команды **iptables -L -nv -t nat**:

```
Chain PREROUTING (policy ACCEPT 211 packets, 16451 bytes)
 pkts bytes target     prot opt in     out     source     destination

Chain POSTROUTING (policy ACCEPT 56 packets, 3686 bytes)
 pkts bytes target     prot opt in     out     source     destination
    2   117 MASQUERADE  all  --  *      eth1    0.0.0.0/0 0.0.0.0/0

Chain OUTPUT (policy ACCEPT 28 packets, 1775 bytes)
 pkts bytes target     prot opt in     out     source     destination
```

В текущей таблице **nat** применен только маскарадинг (специальный тип SNAT) в цепочке POSTROUTING.

Сбор дампа выполнялся с помощью команды **tcpdump -e -i any icmp** на маршрутизаторе **r1**:

```
19:09:52.948349  In ethertype IPv4 (0x0800), length 100:
        10.20.0.2 > 10.10.4.10: ICMP echo request, id 28418, seq 1, length 64
19:09:53.953390  In ethertype IPv4 (0x0800), length 100:
        10.20.0.2 > 10.10.4.10: ICMP echo request, id 28418, seq 2, length 64
19:09:54.952681  In ethertype IPv4 (0x0800), length 100:
        10.20.0.2 > 10.10.4.10: ICMP echo request, id 28418, seq 3, length 64
19:09:55.982291  In ethertype IPv4 (0x0800), length 100:
        10.20.0.2 > 10.10.4.10: ICMP echo request, id 28418, seq 4, length 64
19:09:56.981313  In ethertype IPv4 (0x0800), length 100:
        10.20.0.2 > 10.10.4.10: ICMP echo request, id 28418, seq 5, length 64
```

Как можно видеть сообщения по логам, сообщения ICMP были успешно доставлены до роутера, но дальше они не прошли, так как ответ от хоста **10.10.4.10** не был получен.

4. Проверка трансляции SNAT

NAT - это механизм в сетях TCP/IP, позволяющий преобразовывать IP-адреса транзитных пакетов (проходящих через маршрутизатор и пробрасываемых им далее).

Преобразование адреса методом NAT может производиться почти любым маршрутизирующим устройством - маршрутизатором, сервером доступа, межсетевым экраном. Наиболее популярным является SNAT (Source NAT).

Суть механизма SNAT состоит в замене адреса источника при прохождении пакета в одну сторону и обратной замене адреса назначения в ответном пакете. Наряду с адресами источник/назначение могут также заменяться номера портов источника и назначения.

Для более точной настройки SNAT правило MASQUERADE было заменено настройкой с SNAT.

Для чистоты эксперимента был отключен VPN между двумя маршрутизаторами **r1** и **r2**.

SNAT настраивался на машине **r2** по причине того, что она эмулирует обычный домашний маршрутизатор с доступом в Интернет. В одной сети с данным маршрутизатором находится машина **ws21**, которая эмулирует ПК и которой требуется обеспечить SNAT.

Сценарий iptables на маршрутизаторе **r2**:

```
#!/bin/sh
LAN=eth0
INET=eth1
VPN=tun0

# Удаление всех правил в таблице "filter" (по-умолчанию).
iptables -F

# Удаление правил в таблице "nat" (её надо указать явно).
iptables -F -t nat

# По-умолчанию все маршрутизируемые пакеты выбрасываются.
iptables --policy FORWARD DROP

# ICMP разрешим
iptables -A FORWARD -p icmp -j ACCEPT

# Разрешаем любую маршрутизацию для интерфейса VPN
iptables -A FORWARD -i $VPN -j ACCEPT
iptables -A FORWARD -o $VPN -j ACCEPT

# Включение SNAT для маршрутизируемых пакетов
# адрес назначения исходящего пакета будет заменен на 172.16.1.4
iptables -t nat -A POSTROUTING -j SNAT --to-source 172.16.1.4

# Правила разрешения TCP-сообщений
iptables -A FORWARD -i $INET -o $LAN -p tcp -m conntrack
        --ctstate NEW -j ACCEPT
iptables -A FORWARD -i $INET -o $LAN -m conntrack
        --ctstate ESTABLISHED,RELATED -j ACCEPT
iptables -A FORWARD -i $LAN -o $INET -m conntrack
        --ctstate NEW,ESTABLISHED,RELATED -j ACCEPT
```

Таблица filter на машине **r2**:

```
Chain INPUT (policy ACCEPT 64 packets, 4966 bytes)
 pkts bytes target     prot opt in     out     source       destination

Chain FORWARD (policy DROP 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source       destination
    8   672 ACCEPT     icmp --  *      *       0.0.0.0/0    0.0.0.0/0
    0     0 ACCEPT     all  --  tun0   *       0.0.0.0/0    0.0.0.0/0
    0     0 ACCEPT     all  --  *      tun0    0.0.0.0/0    0.0.0.0/0
    0     0 ACCEPT     tcp  --  eth1   eth0    0.0.0.0/0    0.0.0.0/0
                                               ctstate NEW
    0     0 ACCEPT     all  --  eth1   eth0    0.0.0.0/0    0.0.0.0/0
                                               ctstate RELATED,ESTABLISHED
    0     0 ACCEPT     all  --  eth0   eth1    0.0.0.0/0    0.0.0.0/0
                                               ctstate NEW,RELATED,ESTABLISHED

Chain OUTPUT (policy ACCEPT 67 packets, 5198 bytes)
 pkts bytes target     prot opt in     out     source       destination
```

Таблица nat на машине **r2**:

```
Chain PREROUTING (policy ACCEPT 7 packets, 524 bytes)
 pkts bytes target     prot opt in     out     source        destination

Chain POSTROUTING (policy ACCEPT 13 packets, 822 bytes)
 pkts bytes target     prot opt in     out     source        destination
   16  1148 SNAT       all  --  *      *       0.0.0.0/0     0.0.0.0/0
                                                              to:172.16.1.4

Chain OUTPUT (policy ACCEPT 24 packets, 1550 bytes)
 pkts bytes target     prot opt in     out     source        destination
```

Дамп SNAT в LAN собирался на машине **r2** (**tcpdump -e -i any icmp**):

```
20:50:26.104243  In 10:10:10:10:10:ee (oui Unknown) ethertype IPv4 (0x0800), length 100:
        10.20.0.2 > dns.google: ICMP echo request, id 44802, seq 1, length 64
20:50:26.104271 Out 12:3e:e2:7d:e3:87 (oui Unknown) ethertype IPv4 (0x0800), length 100:
        172.16.1.4 > dns.google: ICMP echo request, id 44802, seq 1, length 64
20:50:26.121197  In c2:f4:23:7f:46:86 (oui Unknown) ethertype IPv4 (0x0800), length 100:
        dns.google > 172.16.1.4: ICMP echo reply, id 44802, seq 1, length 64
20:50:26.121213 Out 3a:40:ee:31:9e:cd (oui Unknown) ethertype IPv4 (0x0800), length 100:
        dns.google > 10.20.0.2: ICMP echo reply, id 44802, seq 1, length 64
```

Дамп SNAT снаружи собирался на хосте, обслуживающем виртуальный машины (**tcpdump -e -i nktapubuntu icmp**):

```
12:50:26.111007 12:3e:e2:7d:e3:87 (oui Unknown) > c2:f4:23:7f:46:86 (oui Unknown),
        ethertype IPv4 (0x0800), length 98: 172.16.1.4 > dns.google: ICMP echo request,
        id 44802, seq 1, length 64
12:50:26.127651 c2:f4:23:7f:46:86 (oui Unknown) > 12:3e:e2:7d:e3:87 (oui Unknown),
        ethertype IPv4 (0x0800), length 98: dns.google > 172.16.1.4: ICMP echo reply,
        id 44802, seq 1, length 64
```

Из вывода видно, что по приходу пакета на NAT-маршрутизатор **r2** от машины **ws1**, NAT-маршрутизатор заменяет адрес источника **10.20.0.2** на свой адрес **172.16.1.4**, а затем пересылает пакет дальше. При получении ответа поведение аналогичное, только NAT-маршрутизатор заменяет IP-адрес назначения с того, который принадлежит маршрутизатору **172.16.1.4**, на тот, который принадлежит машине **10.20.0.2**, ожидающей ответа. Об адресе внутри локальной сети знает только NAT-маршрутизатор.

5. Проверка правил фильтрации

Требовалось провести проверку правил фильтрации с помощью telnet. Для этого на машине **ws21** было организовано подключение к HTTP-серверу, отвечающего за **bmstu.ru**.

Вывод процесса общения telnet на машине **ws21** с HTTP-сервером **bmstu.ru**:

```
Trying 195.19.50.247...
Connected to bmstu.ru.
Escape character is '^]'.
GET
HTTP/1.1 400 Bad Request
Server: nginx/1.12.2
Date: Fri, 11 Dec 2020 21:09:50 GMT
Content-Type: text/html
Content-Length: 173
Connection: close

<html>
<head><title>400 Bad Request</title></head>
<body bgcolor="white">
<center><h1>400 Bad Request</h1></center>
<hr><center>nginx/1.12.2</center>
</body>
</html>
Connection closed by foreign host.
```

Также собирался дамп на маршрутизаторе **r2** с помощью команды **tcpdump -i any tcp**:

```
IP 10.20.0.2.39471 > h247.net50.bmstu.ru.www: S 220020433:220020433(0) win 5840
        <mss 1460,sackOK,timestamp 1276355 0,nop,wscale 1>
IP 172.16.1.4.39471 > h247.net50.bmstu.ru.www: S 220020433:220020433(0) win 5840
        <mss 1460,sackOK,timestamp 1276355 0,nop,wscale 1>
IP h247.net50.bmstu.ru.www > 172.16.1.4.39471: S 1838432673:1838432673(0) ack 220020434
        win 64240 <mss 1460>
IP h247.net50.bmstu.ru.www > 10.20.0.2.39471: S 1838432673:1838432673(0) ack 220020434
        win 64240 <mss 1460>
IP 10.20.0.2.39471 > h247.net50.bmstu.ru.www: . ack 1 win 5840
IP 172.16.1.4.39471 > h247.net50.bmstu.ru.www: . ack 1 win 5840
IP 10.20.0.2.39471 > h247.net50.bmstu.ru.www: P 1:6(5) ack 1 win 5840
IP 172.16.1.4.39471 > h247.net50.bmstu.ru.www: P 1:6(5) ack 1 win 5840
IP h247.net50.bmstu.ru.www > 172.16.1.4.39471: . ack 6 win 64240
IP h247.net50.bmstu.ru.www > 10.20.0.2.39471: . ack 6 win 64240
```

```
IP h247.net50.bmstu.ru.www > 172.16.1.4.39471: FP 1:326(325) ack 6 win 64240
IP h247.net50.bmstu.ru.www > 10.20.0.2.39471: FP 1:326(325) ack 6 win 64240
IP 10.20.0.2.39471 > h247.net50.bmstu.ru.www: F 6:6(0) ack 327 win 6432
IP 172.16.1.4.39471 > h247.net50.bmstu.ru.www: F 6:6(0) ack 327 win 6432
IP h247.net50.bmstu.ru.www > 172.16.1.4.39471: . ack 7 win 64239
IP h247.net50.bmstu.ru.www > 10.20.0.2.39471: . ack 7 win 64239
```

Данный вывод говорит о том, что SNAT и правила фильтрации для TCP настроены корректно.

6. Проверка доступа к внутреннему серверу - DNAT

Трансляция адресов естественным образом блокирует доступ извне ко всем компьютерам внутренней сети, что выглядит преимуществом в случае домашней или офисной сети. Однако, часто нужно организовать доступ к внутреннему серверу при попытке соединения извне. Переброска портов (port forwarding) позволяет обеспечить доступ к внутреннему серверу с "серым"IP-адресом из Интернет. Типичные ее применения: доступ к почтовому или веб-серверу в офисе, доступ к p2p-серверу дома. Решение о переброске входящих пакетов принимается на основе адреса порта назначения и называется DNAT (Destination NAT).

Для проведения экспериментов с настройкой DNAT требовалось отредактировать сценарий iptables на маршрутизаторе **r1**, поскольку данный маршрутизатор находится в сети рядом с машинами, эмулирующими серверы в виртуальной сети, такие как **s11**, **s12**, **s13**.

Было решено поднять один HTTP-сервер **apache2** на машине **s11** с адресом **10.10.4.10** и один почтовый сервер **exim4** на машине **s12** с адресом **10.10.4.20**.

Для того, чтобы серверы были доступны извне сети, был настроен DNAT в iptables и настроена пересылка TCP-сообщений.

Сценарий iptables на маршрутизаторе **r1**:

```
#!/bin/sh
LAN=eth0
INET=eth1
VPN=tun0

# Удаление всех правил в таблице "filter" (по-умолчанию).
iptables -F

# Удаление правил в таблице "nat" (её надо указать явно).
iptables -F -t nat

# По-умолчанию все маршрутизируемые пакеты выбрасываются.
iptables --policy FORWARD DROP

# ICMP разрешим
iptables -A FORWARD -p icmp -j ACCEPT

# Разрешаем любую маршрутизацию для интерфейса VPN
iptables -A FORWARD -i $VPN -j ACCEPT
iptables -A FORWARD -o $VPN -j ACCEPT
```

```
# DNAT для 80 порта и 25
iptables -t nat -A PREROUTING -p tcp --dport 80 -i $INET -j DNAT --to 10.10.4.10:80
iptables -t nat -A PREROUTING -p tcp --dport 25 -i $INET -j DNAT --to 10.10.4.20:25

# TCP разрешаем
iptables -A FORWARD -i $INET -o $LAN -p tcp -m conntrack
        --ctstate NEW -j ACCEPT
iptables -A FORWARD -i $INET -o $LAN -m conntrack
        --ctstate ESTABLISHED,RELATED -j ACCEPT
iptables -A FORWARD -i $LAN -o $INET -m conntrack
        --ctstate NEW,ESTABLISHED,RELATED -j ACCEPT
```

Таблица filter на маршрутизаторе **r1**:

```
Chain INPUT (policy ACCEPT 338 packets, 19614 bytes)
 pkts bytes target     prot opt in     out     source     destination

Chain FORWARD (policy DROP 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source     destination
    0     0 ACCEPT     icmp --  *      *       0.0.0.0/0  0.0.0.0/0
    0     0 ACCEPT     all  --  tun0   *       0.0.0.0/0  0.0.0.0/0
    0     0 ACCEPT     all  --  *      tun0    0.0.0.0/0  0.0.0.0/0
    1    60 ACCEPT     tcp  --  eth1   eth0    0.0.0.0/0  0.0.0.0/0
                                               ctstate NEW
    7  1003 ACCEPT     all  --  eth1   eth0    0.0.0.0/0  0.0.0.0/0
                                               ctstate RELATED,ESTABLISHED
    7  1269 ACCEPT     all  --  eth0   eth1    0.0.0.0/0  0.0.0.0/0
                                               ctstate NEW,RELATED,ESTABLISHED

Chain OUTPUT (policy ACCEPT 342 packets, 19321 bytes)
 pkts bytes target     prot opt in     out     source     destination
```

Таблица nat на маршрутизаторе **r1**:

```
Chain PREROUTING (policy ACCEPT 13 packets, 717 bytes)
 pkts bytes target     prot opt in     out     source    destination
    1    60 DNAT       tcp  --  eth1   *       0.0.0.0/0 0.0.0.0/0
                                               tcp dpt:80 to:10.10.4.10:80
    0     0 DNAT       tcp  --  eth1   *       0.0.0.0/0 0.0.0.0/0
                                               tcp dpt:25 to:10.10.4.20:25

Chain POSTROUTING (policy ACCEPT 23 packets, 1310 bytes)
 pkts bytes target     prot opt in     out     source    destination

Chain OUTPUT (policy ACCEPT 21 packets, 1190 bytes)
 pkts bytes target     prot opt in     out     source    destination
```

Был использован telnet для общения с SMTP-сервером по адресу **172.16.1.3:25**. Также собирался дамп с помощью команды **tcpdump -t -i any tcp** на маршрутизаторе **r1**.

Вывод общения telnet на машине **ws1** с SMTP-сервером по внешнему адресу **172.16.1.3:25**:

```
Trying 172.16.1.3...
Connected to 172.16.1.3.
Escape character is '^]'.
220 s12 ESMTP Exim 4.69 Fri, 11 Dec 2020 21:57:22 +0000
HELO client
250 s12 Hello client [172.16.1.4]
MAIL from:<vladovchinnikov950@gmail.com>
250 OK
RCPT to:<admin@172.16.1.3>
250 Accepted
DATA
354 Enter message, ending with "." on a line by itself
Subject: Hello, world

Hello, world.
.
250 OK id=1knqQ7-0000JT-Ba
```

Часть дампа, собранного с машины **r1**:

```
IP 172.16.1.4.54380 > 172.16.1.3.smtp: S 1961857692:1961857692(0) win 5840
        <mss 1460,sackOK,timestamp 1561264 0,nop,wscale 1>
IP 172.16.1.4.54380 > 10.10.4.20.smtp: S 1961857692:1961857692(0) win 5840
        <mss 1460,sackOK,timestamp 1561264 0,nop,wscale 1>
IP 10.10.4.20.smtp > 172.16.1.4.54380: S 1958695408:1958695408(0) ack 1961857693 win 5792
        <mss 1460,sackOK,timestamp 1561296 1561264,nop,wscale 1>
IP 172.16.1.3.smtp > 172.16.1.4.54380: S 1958695408:1958695408(0) ack 1961857693 win 5792
        <mss 1460,sackOK,timestamp 1561296 1561264,nop,wscale 1>
IP 172.16.1.4.54380 > 172.16.1.3.smtp: . ack 1 win 2920
        <nop,nop,timestamp 1561284 1561296>
IP 172.16.1.4.54380 > 10.10.4.20.smtp: . ack 1 win 2920
        <nop,nop,timestamp 1561284 1561296>
IP 10.10.4.20.46613 > 172.16.1.4.auth: S 1954782312:1954782312(0) win 5840
        <mss 1460,sackOK,timestamp 1561296 0,nop,wscale 1>
IP 10.10.4.20.46613 > 172.16.1.4.auth: S 1954782312:1954782312(0) win 5840
        <mss 1460,sackOK,timestamp 1561296 0,nop,wscale 1>
IP 10.10.4.20.46613 > 172.16.1.4.auth: S 1954782312:1954782312(0) win 5840
        <mss 1460,sackOK,timestamp 1561596 0,nop,wscale 1>
IP 10.10.4.20.46613 > 172.16.1.4.auth: S 1954782312:1954782312(0) win 5840
        <mss 1460,sackOK,timestamp 1561596 0,nop,wscale 1>
IP 10.10.4.20.smtp > 172.16.1.4.54380: P 1:58(57) ack 1 win 2896
        <nop,nop,timestamp 1561799 1561284>
IP 172.16.1.3.smtp > 172.16.1.4.54380: P 1:58(57) ack 1 win 2896
        <nop,nop,timestamp 1561799 1561284>
IP 172.16.1.4.54380 > 172.16.1.3.smtp: . ack 58 win 2920
        <nop,nop,timestamp 1561797 1561799>
IP 172.16.1.4.54380 > 10.10.4.20.smtp: . ack 58 win 2920
        <nop,nop,timestamp 1561797 1561799>
IP 172.16.1.4.54380 > 172.16.1.3.smtp: P 1:14(13) ack 58 win 2920
        <nop,nop,timestamp 1562147 1561799>
IP 172.16.1.4.54380 > 10.10.4.20.smtp: P 1:14(13) ack 58 win 2920
```

```
        <nop,nop,timestamp 1562147 1561799>
IP 10.10.4.20.smtp > 172.16.1.4.54380: . ack 14 win 2896
        <nop,nop,timestamp 1562160 1562147>
IP 172.16.1.3.smtp > 172.16.1.4.54380: . ack 14 win 2896
        <nop,nop,timestamp 1562160 1562147>
```

Был использован веб-браузер для получения веб-страницы по адресу **172.16.1.3:80**, страница ожидаемо содержала текст "It's works". Дамп, перехваченный с помощью команды **tcpdump -t -i any tcp** на маршрутизаторе **r1**:

```
IP 172.16.1.2.53358 > 172.16.1.3.www: S 1951591687:1951591687(0) win 29200
        <mss 1460,sackOK,timestamp 4797927 0,nop,wscale 7>
IP 172.16.1.2.53358 > 10.10.4.10.www: S 1951591687:1951591687(0) win 29200
        <mss 1460,sackOK,timestamp 4797927 0,nop,wscale 7>
IP 10.10.4.10.www > 172.16.1.2.53358: S 570764664:570764664(0) ack 1951591688 win 5792
        <mss 1460,sackOK,timestamp 1443029 4797927,nop,wscale 1>
IP 172.16.1.3.www > 172.16.1.2.53358: S 570764664:570764664(0) ack 1951591688 win 5792
        <mss 1460,sackOK,timestamp 1443029 4797927,nop,wscale 1>
IP 172.16.1.2.53358 > 172.16.1.3.www: . ack 1 win 229
        <nop,nop,timestamp 4797935 1443029>
IP 172.16.1.2.53358 > 10.10.4.10.www: . ack 1 win 229
        <nop,nop,timestamp 4797935 1443029>
IP 172.16.1.2.53358 > 172.16.1.3.www: P 1:330(329) ack 1 win 229
        <nop,nop,timestamp 4797935 1443029>
IP 172.16.1.2.53358 > 10.10.4.10.www: P 1:330(329) ack 1 win 229
        <nop,nop,timestamp 4797935 1443029>
IP 10.10.4.10.www > 172.16.1.2.53358: . ack 330 win 3432
        <nop,nop,timestamp 1443029 4797935>
IP 172.16.1.3.www > 172.16.1.2.53358: . ack 330 win 3432
        <nop,nop,timestamp 1443029 4797935>
IP 10.10.4.10.www > 172.16.1.2.53358: P 1:397(396) ack 330 win 3432
        <nop,nop,timestamp 1443029 4797935>
IP 172.16.1.3.www > 172.16.1.2.53358: P 1:397(396) ack 330 win 3432
        <nop,nop,timestamp 1443029 4797935>
IP 172.16.1.2.53358 > 172.16.1.3.www: . ack 397 win 237
        <nop,nop,timestamp 4797937 1443029>
IP 172.16.1.2.53358 > 10.10.4.10.www: . ack 397 win 237
        <nop,nop,timestamp 4797937 1443029>
IP 172.16.1.2.53358 > 172.16.1.3.www: P 330:640(310) ack 397 win 237
        <nop,nop,timestamp 4798006 1443029>
IP 172.16.1.2.53358 > 10.10.4.10.www: P 330:640(310) ack 397 win 237
        <nop,nop,timestamp 4798006 1443029>
IP 10.10.4.10.www > 172.16.1.2.53358: P 397:898(501) ack 640 win 3968
        <nop,nop,timestamp 1443058 4798006>
IP 172.16.1.3.www > 172.16.1.2.53358: P 397:898(501) ack 640 win 3968
        <nop,nop,timestamp 1443058 4798006>
IP 172.16.1.2.53358 > 172.16.1.3.www: . ack 898 win 245
        <nop,nop,timestamp 4798016 1443058>
IP 172.16.1.2.53358 > 10.10.4.10.www: . ack 898 win 245
        <nop,nop,timestamp 4798016 1443058>
IP 172.16.1.2.53358 > 172.16.1.3.www: . ack 898 win 245
```

```
        <nop,nop,timestamp 4800516 1443058>
IP 172.16.1.2.53358 > 10.10.4.10.www: . ack 898 win 245
        <nop,nop,timestamp 4800516 1443058>
IP 10.10.4.10.www > 172.16.1.2.53358: . ack 640 win 3968
        <nop,nop,timestamp 1444062 4798016>
IP 172.16.1.3.www > 172.16.1.2.53358: . ack 640 win 3968
        <nop,nop,timestamp 1444062 4798016>
IP 10.10.4.10.www > 172.16.1.2.53358: F 898:898(0) ack 640 win 3968
        <nop,nop,timestamp 1444559 4798016>
IP 172.16.1.3.www > 172.16.1.2.53358: F 898:898(0) ack 640 win 3968
        <nop,nop,timestamp 1444559 4798016>
IP 172.16.1.2.53358 > 172.16.1.3.www: F 640:640(0) ack 899 win 245
        <nop,nop,timestamp 4801758 1444559>
IP 172.16.1.2.53358 > 10.10.4.10.www: F 640:640(0) ack 899 win 245
        <nop,nop,timestamp 4801758 1444559>
IP 10.10.4.10.www > 172.16.1.2.53358: . ack 641 win 3968
        <nop,nop,timestamp 1444559 4801758>
IP 172.16.1.3.www > 172.16.1.2.53358: . ack 641 win 3968
        <nop,nop,timestamp 1444559 4801758>
```

Таким образом, все проверки были успешно выполнены, и DNAT работает корректно.