

# Лабораторная работа №2

## Настройка DNS-сервера

Сахно Никита НФИбд-02-23

### Содержание

1	Цель работы .....	1
2	Задание .....	1
3	Выполнение лабораторной работы.....	1
4	Выводы.....	11

### 1 Цель работы

Приобрести практические навыки по установке и конфигурированию DNS-сервера, усвоить принципы работы системы доменных имён.

### 2 Задание

1. Установите на виртуальной машине server DNS-сервер bind и bind-utils.
2. Сконфигурируйте на виртуальной машине server кэширующий DNS-сервер.
3. Сконфигурируйте на виртуальной машине server первичный DNS-сервер.
4. При помощи утилит dig и host проанализируйте работу DNS-сервера.
5. Напишите скрипт для Vagrant, фиксирующий действия по установке и конфигурированию DNS-сервера во внутреннем окружении виртуальной машины server. Соответствующим образом внесите изменения в Vagrantfile

### 3 Выполнение лабораторной работы

Загрузим операционную систему и перейдем в рабочий каталог с проектом: `cd C:\Users\nikita\work\study\nvsakhno\vagrant\` Затем запустим виртуальную машину server с помощью команды: `make server-up` На виртуальной машине server войдем под созданным в предыдущей работе пользователем и откроем терминал. Перейдем в режим суперпользователя и установим bind и bind-utils:

```
$ make server-up
Bringing machine 'server' up with 'virtualbox' provider...
==> server: You assigned a static IP ending in ".1" to this machine.
==> server: This is very often used by the router and can cause the
==> server: network to not work properly. If the network doesn't work
==> server: properly, try changing this IP.
==> server: You assigned a static IP ending in ".1" to this machine.
==> server: This is very often used by the router and can cause the
==> server: network to not work properly. If the network doesn't work
==> server: properly, try changing this IP.
==> server: Clearing any previously set forwarded ports...
==> server: Clearing any previously set network interfaces...
==> server: Preparing network interfaces based on configuration...
server: Adapter 1: nat
server: Adapter 2: intnet
==> server: Forwarding ports...
server: 22 (guest) => 2222 (host) (adapter 1)
==> server: Running 'pre-boot' VM customizations
```

*Команда make server-up*

```
a.net ~]$ sudo -i
# dnf -y install bind bind-utils
Linux 9 - x86_64 12 kB/s | 11 kB 00:00
Linux 9 - x86_64 4.0 MB/s | 20 MB 00:04
```

*Установка bind и bind-utils в режиме суперпользователя*

С помощью утилиты dig сделаем запрос к DNS-адресу [www.yandex.ru](http://www.yandex.ru):

```
; <<>> DiG 9.16.23-RH <<>> www.yandex.ru
;; global options: +cmd
;; Got answer:
;; ->HEADER<<- opcode: QUERY, status: NOERROR, id: 5977
;; flags: qr rd ra; QUERY: 1, ANSWER: 4, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;www.yandex.ru.          IN      A

;; ANSWER SECTION:
www.yandex.ru.          3600    IN      A       77.88.55.88
www.yandex.ru.          3600    IN      A       77.88.55.60
www.yandex.ru.          3600    IN      A       5.255.255.70
www.yandex.ru.          3600    IN      A       5.255.255.77

;; Query time: 10 msec
;; SERVER: 10.0.2.3#53(10.0.2.3)
;; WHEN: Sat Nov 11 14:16:24 UTC 2023
;; MSG SIZE rcvd: 95
```

*Команда dig*

Давайте рассмотрим разделы данного вывода подробнее:

- **HEADER (заголовок):** показывает версию dig, глобальные опции используемые командой и другую дополнительную информацию
- **QUESTION SECTION (секция запроса):** Показывает наш запрос, то есть мы запросили показать A-запись (команда dig без параметров) для домена [www.yandex.ru](http://www.yandex.ru)
- **ANSWER SECTION (секция ответа):** Показывает ответ полученный от DNS, в нашем случае показывает A-запись для [www.yandex.ru](http://www.yandex.ru) Последняя секция это статистика по запросу (служебная информация)- время выполнения запроса (10

мс), имя DNS-сервера который запрашивался, когда был создан запрос и размер сообщения

## Конфигурирование кэширующего DNS-сервера

В отчёте проанализируем построчно содержание файлов `/etc/resolv.conf`, `/etc/named.conf`, `/var/named/named.ca`, `/var/named/named.localhost`, `/var/named/named.loopback`. Рассмотрим `/etc/resolv.conf`. В нём указано имя сервера и его адрес:

```
# Generated by NetworkManager
search dmbelicheva.net
nameserver 10.0.2.3
```

Рассмотрим содержимое файле `/var/named/named.localhost`. В нём есть:

- Запись начала полномочий (SOA), которая указывает начало зоны и включает имя хоста, на котором находится файл данных `name.local`.
- Запись сервера имен (NS), идентифицирующая главный и подчиненные серверы имен DNS.
- Указаны адреса IPv4 и IPv6 локального хоста.

В файле `/var/named/named.loopback` все аналогично, только добавляется:

- PTR-запись для локального хоста

```
$TTL 1D
@      IN SOA  @ rname.invalid. (
                                0      ; serial
                                1D     ; refresh
                                1H     ; retry
                                1W     ; expire
                                3H )   ; minimum

NS     @
A      127.0.0.1
AAAA   ::1
PTR    localhost.
```

### Файлы *loopback* и *localhost*

Далее запустим DNS-сервер, включим запуск DNS-сервера в автозапуск при загрузке системы. Проанализируем отличие в выведенной на экран информации при выполнении команд `dig www.yandex.ru` и `dig @127.0.0.1 www.yandex.ru`:

```

; <<> DiG 9.16.23-RH <<> @127.0.0.1 www.yandex.ru
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->HEADER<- opcode: QUERY, status: NOERROR, id: 54474
;; flags: qr rd ra; QUERY: 1, ANSWER: 4, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: 74db22a239f573ee01000000654f9fbb2cd8ed508af81ef8 (good)
;; QUESTION SECTION:
;www.yandex.ru.                IN      A

;; ANSWER SECTION:
www.yandex.ru.                300     IN      A      5.255.255.70
www.yandex.ru.                300     IN      A      5.255.255.77
www.yandex.ru.                300     IN      A      77.88.55.88
www.yandex.ru.                300     IN      A      77.88.55.60

;; Query time: 653 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Sat Nov 11 15:37:31 UTC 2023
;; MSG SIZE rcvd: 134

```

## Команда dig

При указании опрашиваемого адреса в строке с адресом сервера написан адрес, который указывали, также указаны куки, а время запроса увеличилось.

Сделаем DNS-сервер сервером по умолчанию для хоста server и внутренней виртуальной сети. Для этого требуется изменить настройки сетевого соединения eth0 в NetworkManager, переключив его на работу с внутренней сетью и указав для него в качестве DNS-сервера по умолчанию адрес 127.0.0.1, затем сделаем тоже самое для соединения System eth0. Затем запустим NetworkManager и проверим наличие изменений в файле etc/resolv.conf(адрес сервера изменился на заданный нами):

```

=== nmcli interactive connection editor j===
Editing existing '802-3-ethernet' connection: 'eth0'

Type 'help' or '?' for available commands.
Type 'print' to show all the connection properties.
Type 'describe [

```

## Изменение адреса dns-сервера

Настроим направление DNS-запросов от всех узлов внутренней сети, включая запросы от узла server, через узел server. Для этого внесем изменения в файл /etc/named.conf:

```
root@server:~
GNU nano 5.6.1 /etc/named.conf
// named.conf
//
// Provided by Red Hat bind package to configure the ISC BIND named(8) DNS
// server as a caching only nameserver (as a localhost DNS resolver only).
//
// See /usr/share/doc/bind9/sample/ for example named configuration files.
//
options {
    listen-on port 53 { 127.0.0.1; any; };
    listen-on-v6 port 53 { ::1; };
    directory "/var/named";
    dump-file "/var/named/data/cache_dump.db";
    statistics-file "/var/named/data/named_stats.txt";
    memstatistics-file "/var/named/data/named_mem_stats.txt";
    secroots-file "/var/named/data/named_secroots.txt";
    recursing-file "/var/named/data/named.recursing";
    allow-query { localhost; 192.168.0.0/16; };
};

/*
 * If you are building an AUTHORITATIVE DNS server, do NOT enable recursion.
 * If you are building a RECURSIVE (caching) DNS server, you need to enable
 */
```

## Изменение скрипта

Внесем изменения в настройки межсетевого экрана узла server, разрешив работу с DNS и убедимся, что DNS-запросы идут через узел server, который прослушивает порт 53:

```
lsuf: WARNING: can't stat() fuse.gvfsd-fuse file system /run/user/1001/gvfs
Output information may be incomplete.
avahi-daemon 562 avahi 12u IPv4 18760 0t0 UDP *:mdns
avahi-daemon 562 avahi 13u IPv6 18761 0t0 UDP *:mdns
avahi-daemon 562 avahi 14u IPv4 18762 0t0 UDP *:46522
avahi-daemon 562 avahi 15u IPv6 18763 0t0 UDP *:45440
chronyd 586 chrony 5u IPv4 18658 0t0 UDP localhost:323
chronyd 586 chrony 6u IPv6 18659 0t0 UDP localhost:323
named 8308 named 16u IPv4 56604 0t0 UDP localhost:domain
named 8308 named 19u IPv6 56606 0t0 UDP localhost:domain
named 8308 8309 isc-net-0 named 16u IPv4 56604 0t0 UDP localhost:domain
named 8308 8309 isc-net-0 named 19u IPv6 56606 0t0 UDP localhost:domain
named 8308 8310 isc-timer named 16u IPv4 56604 0t0 UDP localhost:domain
named 8308 8310 isc-timer named 19u IPv6 56606 0t0 UDP localhost:domain
named 8308 8311 isc-socket named 16u IPv4 56604 0t0 UDP localhost:domain
named 8308 8311 isc-socket named 19u IPv6 56606 0t0 UDP localhost:domain
named 8308 8349 isc-net-0 named 16u IPv4 56604 0t0 UDP localhost:domain
named 8308 8349 isc-net-0 named 19u IPv6 56606 0t0 UDP localhost:domain
NetworkManager 8831 root 27u IPv4 66911 0t0 UDP server.dmbelicheva
net:bootpc->_gateway:bootps root 27u IPv4 66911 0t0 UDP server.dmbelicheva
NetworkManager 8831 8837 gmain root 27u IPv4 66911 0t0 UDP server.dmbelicheva
net:bootpc->_gateway:bootps root 27u IPv4 66911 0t0 UDP server.dmbelicheva
NetworkManager 8831 8838 gdbus root 27u IPv4 66911 0t0 UDP server.dmbelicheva
net:bootpc->_gateway:bootps root 27u IPv4 66911 0t0 UDP server.dmbelicheva
```

## Внесение изменений

### Конфигурирование кэширующего DNS-сервера при наличии фильтрации DNS-запросов маршрутизаторами

В случае возникновения в сети ситуации, когда DNS-запросы от сервера фильтруются сетевым оборудованием, следует добавить перенаправление DNS-запросов на конкретный вышестоящий DNS-сервер. Для этого в конфигурационный файл named.conf в секцию options следует добавить:

```
forwarders { список DNS-серверов };
forward first;
```

Текущий список DNS-серверов можно получить, введя на локальном хосте (на котором развёртывается образ виртуальной машины) следующую команду:

```
cat /etc/resolv.conf
```

Мы получили следующие данные для конфигурационного файла named.conf виртуальной машины server:

```
forwarders { 198.168.1.1; };
forward first;
```

```
root@server:~  
GNU nano 5.6.1 /etc/named.conf  
// named.conf  
//  
// Provided by Red Hat bind package to configure the ISC BIND named(8) DNS  
// server as a caching only nameserver (as a localhost DNS resolver only).  
//  
// See /usr/share/doc/bind*/sample/ for example named configuration files.  
//  
options {  
    listen-on port 53 { 127.0.0.1; any; };  
    listen-on-v6 port 53 { ::1; };  
    directory "/var/named";  
    dump-file "/var/named/data/cache_dump.db";  
    statistics-file "/var/named/data/named_stats.txt";  
    memstatistics-file "/var/named/data/named_mem_stats.txt";  
    secroots-file "/var/named/data/named.secroots";  
    recursing-file "/var/named/data/named.recursing";  
    allow-query { localhost; 192.168.0.0/16; };  
    forwarders { 192.168.1.1; }  
    forward first  
  
    /*  
    - If you are building an AUTHORITATIVE DNS server, do NOT enable recursion.  
    - If you are building a RECURSIVE (caching) DNS server, you need to enable  
    recursion.  
    - If your recursive DNS server has a public IP address, you MUST enable access  
    control to limit queries to your legitimate users. Failing to do so will  
    cause your server to become part of large scale DNS amplification  
    attacks. Implementing BCP38 within your network would greatly  
    reduce such attack surface  
    */  
    recursion yes;  
  
    dnssec-enable no;  
    dnssec-validation no;
```

*Изменение скрипта*

## Конфигурирование первичного DNS-сервера

Скопируем шаблон описания DNS-зон named.rfc1912.zones из каталога /etc в каталог /etc/named и переименуем его в eademidova.net:

```
nano /etc/named.conf  
cp /etc/named.rfc1912.zones /etc/named/  
cd /etc/named
```

*Окно терминала*

Включим файл описания зоны /etc/named/nvsakhno.net в конфигурационном файле DNS /etc/named.conf, добавив в нём в конце строку:

```
include "/etc/named/nvsakhno.net";
```

```
root@server:/etc/named
GNU nano 5.6.1 /etc/named.conf

recursing-file "/var/named/data/named.recursing";
allow-query { localhost; 192.168.0.0/16; };
forwarders {192.168.1.1; }
forward first

/*
- If you are building an AUTHORITATIVE DNS server, do NOT enable recursion.
- If you are building a RECURSIVE (caching) DNS server, you need to enable
  recursion.
- If your recursive DNS server has a public IP address, you MUST enable access
  control to limit queries to your legitimate users. Failing to do so will
  cause your server to become part of large scale DNS amplification
  attacks. Implementing BCP38 within your network would greatly
  reduce such attack surface
*/
recursion yes;

dnsssec-enable no;
dnsssec-validation no;

managed-keys-directory "/var/named/dynamic";
geoip-directory "/usr/share/GeoIP";

pid-file "/run/named/named.pid";
session-keyfile "/run/named/session.key";

/* https://fedoraproject.org/wiki/Changes/CryptoPolicy */
include "/etc/crypto-policies/back-ends/bind.config";
};

logging {
    channel default_debug {
        file "data/named.run";
        severity dynamic;
    };
};

zone "." IN {
    type hint;
    file "named.ca";
};

include "/etc/named.rfc1912.zones";
include "/etc/named.root.key";
include "/etc/named/dmbelicheva.net";
```

## Изменение скрипта

Внесём изменения в файл `nvsakhno.net`:

```
// named.rfc1912.zones:
//
// Provided by Red Hat caching-nameserver package
//
// ISC BIND named zone configuration for zones recommended by
// RFC 1912 section 4.1 : localhost TLDs and address zones
// and https://tools.ietf.org/html/rfc6303
// (c)2007 R W Franks
//
// See /usr/share/doc/bind*/sample/ for example named configuration files.
//
// Note: empty-zones-enable yes; option is default.
// If private ranges should be forwarded, add
// disable-empty-zone "."; into options
//
zone "dmbelicheva.net" IN {
    type master;
    file "master/fz/dmbelicheva.net";
    allow-update { none; };
};

zone "1.168.192.in-addr.arpa" IN {
    type master;
    file "master/rz/192.168.1";
    allow-update { none; };
};
```

## Изменение скрипта

В каталоге `/var/named` создадим подкаталоги `master/fz` и `master/rz`, в которых будут располагаться файлы прямой и обратной зоны соответственно, а затем скопируем шаблон прямой DNS-зоны `named.localhost` из каталога `/var/named` в каталог `/var/named/master/fz` и переименуем его в `eademidova.net`:

```
cd /var/named
mkdir -p /var/named/master/fz
mkdir -p /var/named/master/rz
cp /var/named/named.localhost /var/named/master/fz/
cd /var/named/master/fz/
```

### Изменение скрипта

Изменим файл `/var/named/master/fz/user.net`, указав необходимые DNS-записи для прямой зоны:

```
$TTL 1D
@      IN SOA  @ server.dmbelicheva.net. (
                                2023111100    ; serial
                                1D      ; refresh
                                1H      ; retry
                                1W      ; expire
                                3H )   ; minimum

      NS      @
      A       192.168.1.1
$ORIGIN dmbelicheva.net.
server A       192.168.1.1
ns     A       192.168.1.1
```

### Изменение скрипта

Скопируем шаблон обратной DNS-зоны `named.loopback` из каталога `/var/named` в каталог `/var/named/master/rz` и переименуем его в `192.168.1`, а также изменим файл:



```
GNU nano 5.6.1                                192.168.1
$TTL 1D
@      IN SOA  @ server.dmbelicheva.net. (
                                2023111100    ; serial
                                1D      ; refresh
                                1H      ; retry
                                1W      ; expire
                                3H )   ; minimum

      NS      @
      A       192.168.1.1
      PTR     server.dmbelicheva.net.
$ORIGIN 1.168.192.in-addr.arpa.
1      PTR     server.dmbelicheva.net.
1      PTR     ns.dmbelicheva.net.
```

### Изменение скрипта

После изменения доступа к конфигурационным файлам `named` корректно восстановим специальные метки безопасности в SELinux, затем проверим состояние переключателей. В дополнительном терминале запустим в режиме реального времени расширенный лог системных сообщений, чтобы проверить корректность работы системы:



```

Subject: A start job for unit dnf-makecache.service has begun execution
Defined-By: systemd
Support: https://access.redhat.com/support

A start job for unit dnf-makecache.service has begun execution.

The job identifier is 4740.
Nov 11 18:40:14 server.dmbelicheva.net dnf[10949]: Metadata timer caching disabled when running on a battery.
Nov 11 18:40:14 server.dmbelicheva.net systemd[1]: dnf-makecache.service: Deactivated successfully.
Subject: Unit succeeded
Defined-By: systemd
Support: https://access.redhat.com/support

The unit dnf-makecache.service has successfully entered the 'dead' state.
Nov 11 18:40:14 server.dmbelicheva.net systemd[1]: Finished dnf makecache.
Subject: A start job for unit dnf-makecache.service has finished successfully
Defined-By: systemd
Support: https://access.redhat.com/support

A start job for unit dnf-makecache.service has finished successfully.

The job identifier is 4740.

```

## *Запуск расширенного лога системных сообщений*

В случае ошибок перезапустим DNS-сервер:

```

net rz]# systemctl restart named
net rz]#

```

## *Перезапуск сервера*

## **Анализ работы DNS-сервера**

При помощи утилиты dig получим описание DNS-зоны с сервера ns.nvsakhno.net:

```

;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 7956
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: 28d93565e735d0fc01000000654fcb0040c5e70b1bf52b62 (good)
;; QUESTION SECTION:

```

## *Утилита dig*

При помощи утилиты host проанализируем корректность работы DNS-сервера, можно увидеть, что все внесённые нами изменения в работу сервера учтены:

```

127.0.0.1#53 in 8 ms
.net rz]# host -t A dmbelicheva.net
ess 192.168.1.1
.net rz]# host -t PTR 192.168.1.1

```

## *Утилита host*

## **Внесение изменений в настройки внутреннего окружения виртуальной машины**

На виртуальной машине server перейдем в каталог для внесения изменений в настройки внутреннего окружения /vagrant/provision/server/, создадим в нём каталог dns, в который поместим в соответствующие каталоги конфигурационные файлы DNS, а затем в каталоге /vagrant/provision/server создадим исполняемый файл dns.sh:

```

m: name pointer ns.unibetcheva.net.
z]# cd /vagrant
vagrant]# mkdir -p /vagrant/provision/server/dns/etc/named
vagrant]# mkdir -p /vagrant/provision/server/dns/var/named/master/
vagrant]# cp -R /etc/named.conf /vagrant/provision/server/dns/etc/
vagrant]# cp -R /etc/named/* /vagrant/provision/server/dns/etc/named/
vagrant]# cp -R /var/named/master/* /vagrant/provision/server/dns/var/named/master/
vagrant]# cd /vagrant/provision/server
server]# touch dns.sh
server]# chmod +x dns.sh
server]# nano dns.sh

```

*Создание каталога `dns` и перенос в него файлов, создание `dns.sh`*

Запишем в `dns.sh` следующий скрипт:

```

root@server:/vagrant/provision/server
GNU nano 5.6.1 dns.sh
#!/bin/bash
echo "Provisioning script $0"
echo "Install needed packages"
dnf -y install bind bind-utils
echo "Copy configuration files"
cp -R /vagrant/provision/server/dns/etc/* /etc
cp -R /vagrant/provision/server/dns/var/named/* /var/named
chown -R named:named /etc/named
chown -R named:named /var/named
restorecon -vR /etc
restorecon -vR /var/named
echo "Configure firewall"
firewall-cmd --add-service=dns
firewall-cmd --add-service=dns --permanent
echo "Tuning SELinux"
setsebool named_write_master_zones 1
setsebool -P named_write_master_zones 1
echo "Change dns server address"
nmcli connection edit "System eth0" <<EOF
remove ipv4.dns
set ipv4.ignore-auto-dns yes
set ipv4.dns 127.0.0.1
save
quit
EOF
systemctl restart NetworkManager
echo "Start named service"
systemctl enable named
systemctl start named

```

*Изменение скрипта*

Для отработки созданного скрипта во время загрузки виртуальной машины `server` в конфигурационном файле `Vagrantfile` добавим в разделе конфигурации для сервера:

```
server.vm.box = "rocky9"
server.vm.hostname = 'server'

server.vm.boot_timeout = 1440

server.ssh.insert_key = false
server.ssh.username = 'vagrant'
server.ssh.password = 'vagrant'

server.vm.network :private_network,
  ip: "192.168.1.1",
  virtualbox____intnet: true

server.vm.provision "server dummy",
  type: "shell",
  preserve_order: true,
  path: "provision/server/01-dummy.sh"
server.vm.provision "server dns",
  type: "shell",
  preserve_order: true,
  path: "provision/server/dns.sh"
server.vm.provider :virtualbox do |v|
```

*Изменение Vagrantfile*

## 4 Выводы

В процессе выполнения данной лабораторной работы я приобрел практические навыки по установке и конфигурированию DNS-сервера, усвоила принципы работы системы доменных имён.