

Отчет по лабораторной работе №2

Дисциплина: Администрирование сетевых подсистем

Иванов Сергей Владимирович

Содержание

1	Цель работы	4
2	Задание	5
3	Выполнение лабораторной работы	6
3.1	Установка DNS-сервера	6
3.2	Конфигурирование кэширующего DNS-сервера	7
3.2.1	Конфигурирование кэширующего DNS-сервера при отсут- ствии фильтрации DNS-запросов маршрутизаторами . . .	7
3.2.2	Конфигурирование кэширующего DNS-сервера при нали- чии фильтрации DNS-запросов маршрутизаторами	13
4	Ответы на контрольные вопросы	22
5	Выводы	25

Список иллюстраций

3.1	Запуск server	6
3.2	Установка bind	6
3.3	Запрос с помощью утилиты dig	7
3.4	Файл /etc/resolv.conf	8
3.5	Файл /etc/named.conf	8
3.6	Файл /var/named/named.ca	8
3.7	Файл /var/named/named.localhost	9
3.8	Файл /var/named/named.loopback	9
3.9	Запуск DNS-сервера	10
3.10	Команда dig 127.0.0.1 www.yandex.ru	11
3.11	Скрипт маршрутизации	11
3.12	Перезапуск NetworkManager	12
3.13	Редактирование файла /etc/named.conf	12
3.14	Настройки межсетевого экрана server	13
3.15	Проверка что DNS-запросы идут через узел server (порт 53)	13
3.16	Редактирование named.conf	14
3.17	Копирование шаблона описания DNS-зон	14
3.18	Редактирование svivanov.net	14
3.19	Редактирование svivanov.net	16
3.20	Создание подкаталогов	16
3.21	Копирование шаблона прямой DNS-зоны	16
3.22	Редактирование файла /var/named/master/fz/user.net	17
3.23	Копирование шаблона обратной DNS-зоны	17
3.24	Редактирование файла /var/named/master/rz/192.168.1	18
3.25	Испрвление прав доступа	18
3.26	Восстановление меток в SELinux и их рроверка	19
3.27	Перезапуск DNS-сервера	19
3.28	Описание DNS-зоны с сервера ns.svivanov.net	19
3.29	Проверка корректности работы DNS-сервера	20
3.30	Создание каталога dns с нужными файлами	20
3.31	Скрипт dns.sh	21
3.32	Редактирование Vagrantfile	21

1 Цель работы

Целью данной работы является приобретение практических навыков по установке и конфигурированию DNSсервера, усвоение принципов 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

3.1 Установка DNS-сервера

Загрузим операционную систему и перейдем в рабочий каталог с проектом:
`cd /var/tmp/svivanov/vagrant` Запустим виртуальную машину `server`: `vagrant up server` (рис. 1).

```
C:\work_asp\svivanov\vagrant>vagrant up server
Bringing machine 'server' up with 'virtualbox' provider...
==> server: You assigned a static IP ending in ".1" or ":1" to
==> server: This is very often used by the router and can caus
==> server: network to not work properly. If the network doesn
==> server: properly, try changing this IP.
==> server: You assigned a static IP ending in ".1" or ":1" to
```

Рис. 3.1: Запуск server

На виртуальной машине `server` войдем под созданным в предыдущей работе пользователем и откроем терминал. Перейдем в режим суперпользователя: `sudo -i`. Установим `bind` и `bind-utils`: `dnf -y install bind bind-utils` (рис. 2).

```
[svivanov@server.svivanov.net ~]$ sudo -i
[sudo] пароль для svivanov:
[root@server.svivanov.net ~]# dnf install -y bind bind-utils
Last metadata expiration check: 0:00:05 ago on Br 09 сен 2025 09:36:32.
Package bind-utils-32:9.18.33-3.el10.x86_64 is already installed.
Dependencies resolved.
=====
Package                                     Architecture
=====
Installing:
bind                                         x86_64
Installing weak dependencies:
```

Рис. 3.2: Установка bind

В качестве упражнения с помощью утилиты `dig` сделайте запрос, например, к DNSадресу `www.yandex.ru`: `dig www.yandex.ru`. (рис. 3)

```
[root@server.svivanov.net ~]# dig www.yandex.ru

; <<>> DiG 9.18.33 <<>> www.yandex.ru
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 20409
;; flags: qr rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 1410
;; QUESTION SECTION:
;www.yandex.ru.                IN      A

;; ANSWER SECTION:
www.yandex.ru.                22      IN      A      77.88.55.88
www.yandex.ru.                22      IN      A      5.255.255.77
www.yandex.ru.                22      IN      A      77.88.44.55

;; Query time: 53 msec
;; SERVER: 10.0.2.3#53(10.0.2.3) (UDP)
;; WHEN: Tue Sep 09 09:39:47 UTC 2025
;; MSG SIZE rcvd: 90
```

Рис. 3.3: Запрос с помощью утилиты dig

Анализ выведенной информации:

dig - Это инструмент для запроса DNS-серверов. В данном случае она запросила у DNS-сервера IP-адреса, связанные с доменным именем `www.yandex.ru`.

1. Программа отправила запрос на `www.yandex.ru`
2. Результат: успешный ответ с тремя IP-адресами для `www.yandex.ru`:
 - 77.88.55.88
 - 5.255.255.77
 - 77.88.44.55
3. Время выполнения составило 53мс.

3.2 Конфигурирование кэширующего DNS-сервера

3.2.1 Конфигурирование кэширующего DNS-сервера при отсутствии фильтрации DNS-запросов маршрутизаторами

Посмотрим содержание файлов `/etc/resolv.conf`, `/etc/named.conf`, `/var/named/named.ca`, `/var/named/named.localhost` и `/var/named/named.loopback` (рис. 4, 5, 6, 7, 8)

```
# Generated by NetworkManager
search svivanov.net
nameserver 127.0.0.1
~
```

Рис. 3.4: Файл /etc/resolv.conf

```
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 { 127.0.0.1; };
    forward first;

    dnssec-validation no;
    /*
    - 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;

    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 {
```

Рис. 3.5: Файл /etc/named.conf

```
;
; FORMERLY NS.INTERNIC.NET
;
.                3600000      NS      A.ROOT-SERVERS.NET.
A.ROOT-SERVERS.NET. 3600000      A      198.41.0.4
A.ROOT-SERVERS.NET. 3600000      AAAA    2001:503:ba3e::2:30
;
; FORMERLY NS1.ISI.EDU
;
.                3600000      NS      B.ROOT-SERVERS.NET.
B.ROOT-SERVERS.NET. 3600000      A      170.247.170.2
B.ROOT-SERVERS.NET. 3600000      AAAA    2801:1b8:10::b
;
; FORMERLY C.PSI.NET
;
.                3600000      NS      C.ROOT-SERVERS.NET.
C.ROOT-SERVERS.NET. 3600000      A      192.33.4.12
C.ROOT-SERVERS.NET. 3600000      AAAA    2001:500:2::c
;
; FORMERLY TERP.UMD.EDU
```

Рис. 3.6: Файл /var/named/named.ca


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

    NS      @
    A       127.0.0.1
    AAAA    ::1
```

Рис. 3.7: Файл /var/named/named.localhost

```
$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.
```

Рис. 3.8: Файл /var/named/named.loopback

Анализ файла */etc/resolv.conf* (уже отредатирован):

- Конфигурация DNS-клиента
- Указан DNS-сервер: 127.0.0.1
- Система использует публичные DNS-серверы

Анализ файла */etc/named.conf*:

- Основной конфигурационный файл BIND (DNS-сервер)
- Сервер настроен только для localhost (127.0.0.1)
- DNS-сервер работает только для локальных запросов

Анализ файла */var/named/named.ca*:

- Корневые DNS-серверы интернета
- Список root-серверов (A.ROOT-SERVERS.NET и т.д.)
- Кэш корневых серверов для работы DNS

Анализ файла */var/named/named.localhost:*

- Зона localhost для прямых запросов
- Настройки зоны для localhost (127.0.0.1)
- Базовая конфигурация для локальной зоны

Анализ файла */var/named/named.loopback:*

- Зона обратных запросов для localhost
- Обратная зона для 127.0.0.1
- Настройки reverse DNS для локальной сети

Запускаем DNS-сервер: `systemctl start named`. Включим запуск DNS-сервера в автозапуск при загрузке системы: `systemctl enable named` (рис. 9)

```
[root@server.svivanov.net ~]# systemctl start named
[root@server.svivanov.net ~]# systemctl enable named
Created symlink '/etc/systemd/system/multi-user.target.wants/named.service' ->
[root@server.svivanov.net ~]#
```

Рис. 3.9: Запуск DNS-сервера

Теперь выполним команду `dig [127.0.0.1?] www.yandex.ru`. При выполнении команды `dig www.yandex.ru` 1.1.1.1 (публичный DNS Cloudflare) - использовался автоматически, а при выполнении команды `dig [127.0.0.1?] www.yandex.ru` 127.0.0.1 (локальный DNS-сервер) - указан явно. (рис. 10)

```
[root@server.svivanov.net ~]# dig @127.0.0.1 www.yandex.ru
;; communications error to 127.0.0.1#53: timed out

; <<>> DiG 9.18.33 <<>> @127.0.0.1 www.yandex.ru
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 62030
;; flags: qr rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: 9a9dc3a299db86b20100000068bff6ca6ca9d9c2a3ca1d17 (good)
;; QUESTION SECTION:
;www.yandex.ru.                IN      A

;; ANSWER SECTION:
www.yandex.ru.                600     IN      A      77.88.55.88
www.yandex.ru.                600     IN      A      5.255.255.77
www.yandex.ru.                600     IN      A      77.88.44.55

;; Query time: 3693 msec
;; SERVER: 127.0.0.1#53(127.0.0.1) (UDP)
;; WHEN: Tue Sep 09 09:43:38 UTC 2025
;; MSG SIZE rcvd: 118
```

Рис. 3.10: Команда dig 127.0.0.1 www.yandex.ru

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:

```
nmcli connection edit eth0
remove ipv4.dns
set ipv4.ignore-auto-dns yes
set ipv4.dns 127.0.0.1
save
quit (рис. 11)
```

```
[root@server.svivanov.net ~]# nmcli connection edit eth0

===| интерактивный редактор подключений nmcli |===

Редактируется существующее подключение «802-3-ethernet»: «eth0»

Для просмотра доступных команд введите «help» или «?».
Чтобы просмотреть все свойства подключения введите «print».
Для просмотра описания свойства введите «describe [<параметр>.<свойство>]».

Возможно изменить следующие параметры: connection, 802-3-ethernet (ethernet), 802-1x, dcb, sriov,
, hostname, link, tc, proxy
nmcli> remove ipv4.dns
nmcli> set ipv4.ignore-auto-dns yes
nmcli> set ipv4.dns 127.0.0.1
nmcli> save
Подключение «eth0» (25c2ca05-60fa-4af7-8fa6-4f501be849c5) успешно обновлено.
nmcli> quit
[root@server.svivanov.net ~]#
```

Рис. 3.11: Скрипт маршрутизации

Перезапустим NetworkManager: `systemctl restart NetworkManager`. Проверим наличие изменений в файле `/etc/resolv.conf`. (рис. 12)

```
[root@server.svivanov.net ~]# systemctl restart NetworkManager
[root@server.svivanov.net ~]# cd /etc
[root@server.svivanov.net etc]# cat resolv.conf
# Generated by NetworkManager
search svivanov.net
nameserver 127.0.0.1
[root@server.svivanov.net etc]#
```

Рис. 3.12: Перезапуск NetworkManager

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`, заменив строку

`listen-on port 53 { 127.0.0.1; };`

на

`listen-on port 53 { 127.0.0.1; any; };`

и строку

`allow-query { localhost; };`

на

`allow-query { localhost; 192.168.0.0/16; };` (рис. 13)

```
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; ];
```

Рис. 3.13: Редактирование файла `/etc/named.conf`

Внесем изменения в настройки межсетевого экрана узла `server`, разрешив работу с DNS: (рис. 14)

`firewall-cmd --add-service=dns`

`firewall-cmd --add-service=dns --permanent`

```
[root@server.svivanov.net etc]# firewall-cmd --add-service=dns
success
[root@server.svivanov.net etc]# firewall-cmd --add-service=dns --permanent
success
[root@server.svivanov.net etc]#
```

Рис. 3.14: Настройки межсетевого экрана server

Убедимся, что DNS-запросы идут через узел server, который прослушивает порт 53. Для этого используем команду `lsof: lsof | grep UDP` (рис. 15)

```
[root@server.svivanov.net etc]# lsof | grep UDP
lsof: WARNING: can't stat() fuse.gvfsd-fuse file system /run/user/1001/gvfs
Output information may be incomplete.
lsof: WARNING: can't stat() fuse.portal file system /run/user/1001/doc
Output information may be incomplete.
avahi-daemon 856      avahi  12u    IPv4        9587      0t0      UDP *:mdns
avahi-daemon 856      avahi  13u    IPv6        9588      0t0      UDP *:mdns
chronyd      882      chrony 5u     IPv4       10280     0t0      UDP localhost:323
chronyd      882      chrony 6u     IPv6       10281     0t0      UDP localhost:323
named        14756    named  41u    IPv4       69969     0t0      UDP localhost:domain
named        14756    named  42u    IPv4       69970     0t0      UDP localhost:domain
named        14756    named  43u    IPv4       69971     0t0      UDP localhost:domain
named        14756    named  44u    IPv4       69972     0t0      UDP localhost:domain
named        14756    named  53u    IPv6       69977     0t0      UDP localhost:domain
named        14756    named  54u    IPv6       69978     0t0      UDP localhost:domain
named        14756    named  55u    IPv6       69979     0t0      UDP localhost:domain
named        14756    named  56u    IPv6       69980     0t0      UDP localhost:domain
named        14756 14757 isc-net-0 named  41u    IPv4       69969     0t0      UDP localhost:domain
named        14756 14757 isc-net-0 named  42u    IPv4       69970     0t0      UDP localhost:domain
named        14756 14757 isc-net-0 named  43u    IPv4       69971     0t0      UDP localhost:domain
```

Рис. 3.15: Проверка что DNS-запросы идут через узел server (порт 53)

3.2.2 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; (рис. 16)
```

Кроме того, возможно вышестоящий DNS-сервер может не поддерживать технологию DNSSEC, тогда следует в конфигурационном файле `named.conf` указать следующие настройки:

```
dnssec-enable no;
dnssec-validation no;
```

```
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 {127.0.0.1};
    forward first;

    dnssec-enable no;
    dnssec-validation no;
}
```

Рис. 3.16: Редактирование named.conf

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

```
cp /etc/named.rfc1912.zones /etc/named/
```

```
cd /etc/named
```

```
mv /etc/named/named.rfc1912.zones /etc/named/svivanov.net (рис. 17)
```

```
[root@server.svivanov.net ~]# cp /etc/named.rfc1912.zones /etc/named/
[root@server.svivanov.net ~]# cd /etc/named
[root@server.svivanov.net named]# mv /etc/named/named.rfc1912.zones /etc/named/user.net
[root@server.svivanov.net named]#
```

Рис. 3.17: Копирование шаблона описания DNS-зон

Включим файл описания зоны /etc/named/svivanov.net в конфигурационном файле DNS /etc/named.conf, добавив в нём в конце строку: include “/etc/named/svivanov.net”; (рис. 18)

```
include "/etc/named/svivanov.net";
include "/etc/named.rfc1912.zones";
include "/etc/named.root.key";
"/etc/named.conf" 60L, 1816B
```

Рис. 3.18: Редактирование svivanov.net

Откроем файл /etc/named/svivanov.net на редактирование и вместо зоны zone “localhost.localdomain” IN {
type master;

```
file "named.localhost";  
allow-update { none; };  
};
```

пропишем свою прямую зону:

```
zone "user.net" IN {  
type master;  
file "master/fz/user.net";  
allow-update { none; };  
};
```

Далее, вместо зоны

```
zone "1.0.0.127.in-addr.arpa" IN {  
type master;  
file "named.loopback";  
allow-update { none; };  
};
```

пропишем свою обратную зону:

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

Остальные записи в файле /etc/named/user.net удалим. (рис. 19)

```

zone "localhost.localdomain" IN {
    type primary;
    file "master/fz/user.net";
    allow-update { none; };
};

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

```

Рис. 3.19: Редактирование svivanov.net

В каталоге /var/named создадим подкаталоги master/fz и master/rz, в которых будут располагаться файлы прямой и обратной зоны соответственно:

```
cd /var/named
```

```
mkdir -p /var/named/master/fz
```

```
mkdir -p /var/named/master/rz (рис. 20)
```

```

[root@server.svivanov.net named]# mkdir -p /var/named/master/fz
[root@server.svivanov.net named]# mkdir -p /var/named/master/rz
[root@server.svivanov.net named]#

```

Рис. 3.20: Создание подкаталогов

Скопируем шаблон прямой DNS-зоны named.localhost из каталога /var/named в каталог /var/named/master/fz и переименуем его в user.net:

```
cp /var/named/named.localhost /var/named/master/fz/
```

```
cd /var/named/master/fz/
```

```
mv named.localhost user.net (рис. 21)
```

```

[root@server.svivanov.net ~]# cp /var/named/named.localhost /var/named/master/fz/
[root@server.svivanov.net ~]# cd /var/named/master/fz/
[root@server.svivanov.net fz]# mv named.localhost user.net
[root@server.svivanov.net fz]# mv named.localhost svivanov.net
mv: cannot stat 'named.localhost': Нет такого файла или каталога
[root@server.svivanov.net fz]# mv user.net svivanov.net
[root@server.svivanov.net fz]#

```

Рис. 3.21: Копирование шаблона прямой DNS-зоны

Изменим файл /var/named/master/fz/user.net, указав необходимые DNS-записи для прямой зоны. В этом файле DNS-имя сервера @ rname.invalid.

должно быть заменено на @ server.user.net.; формат серийного номера ГГГГ-ММДДВВ (ГГГГ — год, ММ — месяц, ДД — день, ВВ — номер ревизии); адрес в А-записи должен быть заменён с 127.0.0.1 на 192.168.1.1; в директиве \$ORIGIN должно быть задано текущее имя домена user.net. (вместо user должен быть указан ваш логин), а затем указаны имена и адреса серверов в этом домене в виде А-записей DNS (на данном этапе должен быть прописан сервер с именем ns и адресом 192.168.1.1). (рис. 22)

```
$TTL 1D
@      IN SOA  server.svivanov.net. (
                                2025090900      ; serial
                                1D                ; refresh
                                1H                ; retry
                                1W                ; expire
                                3H )              ; minimum

      NS   server.svivanov.net.
      A    192.168.1.1
$ORIGIN svivanov.net.
server A   192.168.1.1
ns      A   192.168.1.1
```

Рис. 3.22: Редактирование файла /var/named/master/fz/user.net

Скопируем шаблон обратной DNS-зоны named.loopback из каталога /var/named в каталог /var/named/master/rz и переименуем его в 192.168.1:

```
cp /var/named/named.loopback /var/named/master/rz/
cd /var/named/master/rz/
mv named.loopback 192.168.1 (рис. 23)
```

```
[root@server.svivanov.net fz]# vim svivanov.net
[root@server.svivanov.net fz]# cp /var/named/named.loopback /var/named/master/rz/
[root@server.svivanov.net fz]# cd /var/named/master/rz
[root@server.svivanov.net rz]# mv named.loopback 192.168.1\
```

Рис. 3.23: Копирование шаблона обратной DNS-зоны

Изменим файл /var/named/master/rz/192.168.1, указав необходимые DNS-записи для обратной зоны. В этом файле DNS-имя сервера @ rname.invalid. должно быть заменено на @ server.user.net.; формат серийного номера ГГГГ-ММДДВВ (ГГГГ — год, ММ — месяц, ДД — день, ВВ — номер ревизии); адрес в А-записи должен быть заменён с 127.0.0.1 на 192.168.1.1; в директиве \$ORIGIN

должно быть задано название обратной зоны в виде 1.168.192.in-addr.arpa., затем заданы PTR-записи. (рис. 24)

```
$TTL 1D
@      IN SOA  server.user.net. (
                                2025090900 ; serial
                                1D      ; refresh
                                1H      ; retry
                                1W      ; expire
                                3H )    ; minimum

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

Рис. 3.24: Редактирование файла /var/named/master/rz/192.168.1

Далее требуется исправить права доступа к файлам в каталогах /etc/named и /var/named, чтобы демон named мог с ними работать:

```
chown -R named:named /etc/named
```

```
chown -R named:named /var/named (рис. 25)
```

```
[root@server.svivanov.net rz]# chown -R named:named /etc/named
[root@server.svivanov.net rz]# chown -R named:named /var/named
[root@server.svivanov.net rz]#
```

Рис. 3.25: Испрвление прав доступа

В системах с запущенным SELinux все процессы и файлы имеют специальные метки безопасности, используемые системой для принятия решений по доступу к этим процессам и файлам. После изменения доступа к конфигурационным файлам named требуется корректно восстановить их метки в SELinux:

```
restorecon -vR /etc
```

```
restorecon -vR /var/named
```

Для проверки состояния переключателей SELinux, относящихся к named, введем: `getsebool -a | grep named`. Дадим named разрешение на запись в файлы DNS-зоны:

```
setsebool named_write_master_zones 1
```

```
setsebool -P named_write_master_zones 1 (рис. 26)
```

```
[root@server.svivanov.net rz]# restorecon -vR /etc
Relabeled /etc/NetworkManager/system-connections/eth1.nmconnection from unconfined.
object_r:NetworkManager_etc_rw_t:s0
[root@server.svivanov.net rz]# restorecon -vR /var/named
[root@server.svivanov.net rz]# getsebool -a | grep named
named_tcp_bind_http_port --> off
named_write_master_zones --> on
[root@server.svivanov.net rz]# setsebool named_write_master_zones 1
[root@server.svivanov.net rz]# setsebool -P named_write_master_zones 1
[root@server.svivanov.net rz]#
```

Рис. 3.26: Восстановление меток в SELinux и их проверка

В дополнительном терминале запустим в режиме реального времени расширенный лог системных сообщений, чтобы проверить корректность работы системы: `journalctl -x -f`. В первом терминале перезапустим DNS-сервер: `systemctl restart named` (рис. 27)

```
[4]+ Stopped systemctl status named.service
[root@server.svivanov.net etc]# vim named.conf
[root@server.svivanov.net etc]# systemctl restart named
[root@server.svivanov.net etc]#
```

Рис. 3.27: Перезапуск DNS-сервера

При помощи утилиты `dig` получим описание DNS-зоны с сервера `ns.svivanov.net`: `dig ns.user.net` (рис. 28)

```
[root@server.svivanov.net etc]# dig ns.svivanov.net

; <<>> DiG 9.18.33 <<>> ns.svivanov.net
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 56402
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: d84a114d7912ea7e0100000068c020611e2f28d263ad1e58 (good)
;; QUESTION SECTION:
ns.svivanov.net.                IN      A

;; ANSWER SECTION:
ns.svivanov.net.                86400   IN      A      192.168.1.1
```

Рис. 3.28: Описание DNS-зоны с сервера ns.svivanov.net

Анализ выведенной информации: Команда `dig ns.svivanov.net` запросила IP-адрес DNS-сервера домена.

- Сервер: 127.0.0.1 (локальный)

- Ответ: ns.svivanov.net = 192.168.1.1
- Статус: aa (authoritative answer) - ответ авторитативный
- Время: 1 мс

Локальный DNS-сервер отвечает, что DNS-сервер домена находится по адресу 192.168.1.1.

При помощи утилиты `host` проанализируем корректность работы DNS-сервера:

```
host -l user.net
```

```
host -a user.net
```

```
host -t A user.net
```

```
host -t PTR 192.168.1.1
```

Как видим, сервер работает корректно. (рис. 29)

```
[root@server.svivanov.net etc]# host -t A svivanov.net
svivanov.net has address 192.168.1.1
[root@server.svivanov.net etc]# host -t PTR 192.168.1.1 svivanov.net
Using domain server:
Name: svivanov.net
Address: 192.168.1.1#53
Aliases:

1.1.168.192.in-addr.arpa domain name pointer ns.svivanov.net.
1.1.168.192.in-addr.arpa domain name pointer server.svivanov.net.
[root@server.svivanov.net etc]#
```

Рис. 3.29: Проверка корректности работы DNS-сервера

На виртуальной машине `server` перейдем в каталог для внесения изменений в настройки внутреннего окружения `/vagrant/provision/server/`, создайте в нём каталог `dns`, в который поместим в соответствующие каталоги конфигурационные файлы DNS: (рис. 30)

```
[root@server.svivanov.net etc]# cd /vagrant
[root@server.svivanov.net vagrant]# mkdir -p /vagrant/provision/server/dns/etc/named
[root@server.svivanov.net vagrant]# mkdir -p /vagrant/provision/server/dns/var/named/master
[root@server.svivanov.net vagrant]# cp -R /etc/named.conf/vagrant/provision/server/dns/etc/
cp: missing destination file operand after '/etc/named.conf/vagrant/provision/server/dns/etc/'
Try 'cp --help' for more information.
[root@server.svivanov.net vagrant]# cp -R /etc/named.conf/vagrant/provision/server/dns/etc
cp: missing destination file operand after '/etc/named.conf/vagrant/provision/server/dns/etc'
Try 'cp --help' for more information.
[root@server.svivanov.net vagrant]# cp -R /etc/named.conf /vagrant/provision/server/dns/etc
[root@server.svivanov.net vagrant]# cp -R /etc/named/* /vagrant/provision/server/dns/etc/named/
[root@server.svivanov.net vagrant]# cp -R /var/named/master/* /vagrant/provision/server/dns/var/named/master/
[root@server.svivanov.net vagrant]#
```

Рис. 3.30: Создание каталога `dns` с нужными файлами

В каталоге /vagrant/provision/server создадим исполняемый файл dns.sh:

```
touch dns.sh
```

```
chmod +x dns.sh
```

Открыв его на редактирование, пропишем в нём следующий скрипт: (рис. 31)

```
#!/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"
```

Рис. 3.31: Скрипт dns.sh

Для отработки созданного скрипта во время загрузки виртуальной машины server в конфигурационном файле Vagrantfile добавим в разделе конфигурации для сервера: (рис. 32)

```
server.vm.provision "server dns",
    type: "shell",
    preserve_order: true,
    path: "provision/server/dns.sh"
```

Рис. 3.32: Редактирование Vagrantfile

4 Ответы на контрольные вопросы

1. Что такое DNS?

Это система, предназначенная для преобразования человекочитаемых доменных имен в IP-адреса, используемые компьютерами для распознавания друг друга в сети.

2. Каково назначение кэширующего DNS-сервера?

Его главная цель — ускорить получение ответа и снизить нагрузку на сеть. Он запоминает результаты предыдущих запросов на некоторое время. Если два разных пользователя запрашивают один и тот же сайт, второму ответ придет мгновенно из кэша, без повторного опроса внешних серверов.

3. Чем отличается прямая DNS-зона от обратной?

Прямая зона преобразует доменные имена в IP-адреса, обратная зона выполняет обратное: преобразует IP-адреса в доменные имена.

4. В каких каталогах и файлах располагаются настройки DNS-сервера?

Кратко охарактеризуйте, за что они отвечают.

В Linux-системах обычно используется файл `/etc/named.conf` для общих настроек. Зоны хранятся в файлах в каталоге `/var/named/`, например, `/var/named/example.com.zone`

5. Что указывается в файле `resolv.conf`?

В этом файле прописываются адреса DNS-серверов, которые будет использовать эта машина для преобразования имен.

6. Какие типы записи описания ресурсов есть в DNS и для чего они используются?

A (IPv4-адрес), AAAA (IPv6-адрес), CNAME (каноническое имя), MX (почтовый сервер), NS (имя сервера), PTR (обратная запись), SOA (начальная запись зоны), TXT (текстовая информация).

7. Для чего используется домен in-addr.arpa?

Используется для обратного маппинга IP-адресов в доменные имена.

8. Для чего нужен демон named?

Это DNS-сервер, реализация BIND (Berkeley Internet Name Domain).

9. В чём заключаются основные функции slave-сервера и master-сервера?

Master-сервер хранит оригинальные записи зоны, slave-серверы получают копии данных от master-сервера

10. Какие параметры отвечают за время обновления зоны?

refresh, retry, expire, и minimum.

11. Как обеспечить защиту зоны от скачивания и просмотра?

Можно запретить трансфер зоны (операцию zone transfer) для посторонних серверов, разрешив его только для своих слейвов с помощью директивы allow-transfer в named.conf.

12. Какая запись RR применяется при создании почтовых серверов?

MX (Mail Exchange).

13. Как протестировать работу сервера доменных имён?

Использовать команды nslookup, dig, или host.

14. Как запустить, перезапустить или остановить какую-либо службу в системе?

systemctl start|stop|restart .

15. Как посмотреть отладочную информацию при запуске какого-либо сервиса или службы?

Использовать опции, такие как -d или -v при запуске службы.

16. Где храниться отладочная информация по работе системы и служб? Как её посмотреть?

В системных журналах, доступных через journalctl

17. Как посмотреть, какие файлы использует в своей работе тот или иной процесс?

- lsof -p или fuser -v

18. Приведите несколько примеров по изменению сетевого соединения при помощи командного интерфейса nmcli.

nmcli connection up|down .

19. Что такое SELinux?

Это мандатный контроль доступа для ядра Linux.

20. Что такое контекст (метка) SELinux?

Метка, определяющая, какие ресурсы могут быть доступны процессу или объекту.

21. Как восстановить контекст SELinux после внесения изменений в конфигурационные файлы?

restorecon -Rv .

22. Как создать разрешающие правила политики SELinux из файлов журналов, содержащих сообщения о запрете операций?

audit2allow.

23. Что такое булевый переключатель в SELinux?

Это параметр, который включает или отключает определенные аспекты защиты SELinux.

24. Как посмотреть список переключателей SELinux и их состояние?

getsebool -a.

25. Как изменить значение переключателя SELinux?

setsebool -P <on|off>.

5 Выводы

В ходе выполнения лабораторной работы мы приобрели практические навыки по установке и конфигурированию DNSсервера, а также усвоили принципы работы системы доменных имён.