

## Лабораторная работа № 2. Настройка DNS-сервера

### 2.1. Цель работы

Приобретение практических навыков по установке и конфигурированию DNS-сервера, усвоение принципов работы системы доменных имён.

### 2.2. Предварительные сведения

#### 2.2.1. Основные понятия DNS

*Система доменных имён (Domain Name System, DNS)* — распределённая система (распределённая база данных), ставящая в соответствие доменному имени хоста (компьютера или другого сетевого устройства) IP-адрес, и наоборот.

*DNS-сервер* — специализированное программное обеспечение для обслуживания DNS.

*DNS-клиент* — специализированная библиотека (или программа) для работы с DNS.

В качестве серверов доменных имён чаще всего используются различные версии BIND (Berkeley Internet Name Domain), <http://www.isc.org/software/bind>.

*Зона* — логический узел в дереве имён.

*Домен* — название зоны в системе доменных имён сети «Интернет». Структура доменного имени отражает порядок следования зон в иерархическом виде.

*Поддомен (subdomain)* — имя подчинённой зоны.

Спецификация DNS определяет следующие **типы DNS-серверов**:

- *первичный мастер-сервер (primary master)* — производит загрузку данных для зоны из файла на машине-сервере;
- *дополнительный, или вторичный, мастер-сервер (secondary master)* — получает данные зоны от другого DNS-сервера, называемого его *мастером (master server)*;
- *кэшующий* — получает рекурсивные запросы от клиентов и выполняет их с помощью нерекурсивных запросов к авторитативным серверам.

*Файлы данных зоны* — файлы, из которых первичные DNS-серверы производят чтение зональных данных. Вторичные DNS-серверы также могут загружать зональные данные из файлов.

**Директивы управления:**

- директива \$ORIGIN определяет текущее имя домена;
- директива \$INCLUDE используется для того, чтобы в файл описания зоны можно было включить содержание другого файла.

Формат записи:

```
[<comment>]
$ORIGIN [<comment>]
$INCLUDE [] [<comment>]
```

В квадратные скобки [ ] заключены необязательные параметры, а в угловые скобки < > — сущности.

**RR-записи** описывают все узлы сети в зоне и помечают делегирование поддоменов.

**Типы записи описания ресурсов:**

- SOA-запись — указывает на авторитативность для зоны;
- NS-запись — перечисляет DNS-серверы зоны;
- A — задаёт отображение имени узла в IP-адрес;
- PTR — задаёт отображение IP-адреса в имя узла;
- CNAME — задаёт каноническое имя (для псевдонимов);
- MX — задаёт имена почтовых серверам.

**Формат записи SOA:**

```
[zone] [ttl] IN SOA origin contact (
                                serial refresh retry expire minimum)
```

- *zone* — имя зоны;

- *ttl* — время кэширования (в SOA всегда пустое, определяется директивой управления \$TTL);
- *IN* — класс данных Internet;
- *origin* — доменное имя primary master сервера зоны;
- *contact* — почтовый адрес лица, осуществляющего администрирование зоны (так как символ @ имеет особый смысл при описании зоны, то вместо него в почтовом адресе используется символ «.»);
- *serial* — серийный номер файла зоны в нотации ГТГГММДДВВ (учёт изменений файла описания зоны);
- *refresh* — интервал времени, после которого slave-сервер обязан обратиться к master-серверу с запросом на верификацию своего описания зоны;
- *retry* — интервал времени, после которого slave-сервер должен повторить попытку синхронизировать описание зоны с master сервером;
- *expire* — интервал времени, после которого slave-сервер должен прекратить обслуживание запросов к зоне, если он не смог в течение этого времени верифицировать описание зоны, используя информацию с master сервера;
- *minimum* — время негативного кэширования (negative caching), т.е. время кэширования ответов, которые утверждают, что установить соответствие между доменным именем и IP-адресом нельзя.

**Формат записи NS:**

```
[domain][ttl] IN NS [server]
```

Здесь *domain* — имя домена, для которого сервер, указанный последним аргументом записи NS, поддерживает описание зоны; *server* — доменное имя сервера.

**Формат адресной записи:**

```
[host][ttl] IN A [address]
```

Здесь *host* — доменное имя хоста; *address* — IP-адрес машины.

**Формат PTR-записи имеет следующий вид:**

```
[name][ttl] IN PTR [host]
```

Здесь *name* — номер (не реальный IP-адрес машины, а имя в специальном домене in-addr.arpa или в одной из его зон); *host* — доменное имя хоста.

**Формат MX-записи:**

```
[name] [ttl] IN MX [preference] [host]
```

Здесь *name* — имя машины или домена, на который может отправляться почта; *preference* — приоритет почтового сервера, имя которого (поле *host*) указано последним аргументом в поле данных MX-записи.

**Формат записи CNAME:**

```
[nickname] [ttl] IN CNAME [host]
```

Здесь поле *nickname* определяет синоним для канонического имени, которое задаётся в поле *host*.

## 2.2.2. Сетевые утилиты диагностики DNS

### 2.2.2.1. Утилита dig

Утилита dig (domain information groper) предоставляет пользователю интерфейс командной строки для обращения к системе DNS, позволяет формировать запросы о доменах DNS-серверам. Утилита dig входит в стандартный комплект DNS сервера BIND.

**Формат команды dig:**

```
dig [@server] domain [query-type] [query-class]  
[+query-option] [-dig-option] [%comment]
```

Здесь *server* — имя DNS-сервера. В качестве имени можно указать как имя хоста, так и его IP-адрес.

Параметр *query-type* — тип исходной RR-записи, который можно указать в запросе (A, SOA, NS и MX). Для получения всей информации о домене можно указать *query-type any*.

Параметр *query-class* — класс сетевой информации, который также можно указывать в запросе. По умолчанию этот параметр всегда будет IN для сети Internet.

Параметр `+query-option` используется для изменения значения параметра в пакете DNS или изменения формата вывода результатов работы `dig`.

Более подробную информацию по работе с утилитой `dig` можно найти в руководстве `man`.

### 2.2.2.2. Утилита `host`

Утилита `host` предназначена для выполнения запросов к DNS-серверам.

**Формат команды `host`:**

```
host [-l] [-v] [-w] [-r] [-d] [-t querytype]
    [-a] host [server]
```

Здесь `-l` — выводит полную информацию о домене, `-v` — использует подробный формат при выводе результатов, `-w` — заставляет команду `host` ожидать ответа, `-r` — включает режим рекурсии, `-d` — включает режим отладки, `-t querytype` — определяет тип запроса, `-a` — восстанавливает все записи в DNS.

## 2.3. Задание

1. Установите на виртуальной машине `server` DNS-сервер `bind` и `bind-utils` (см. раздел 2.4.1).
2. Сконфигурируйте на виртуальной машине `server` кэширующий DNS-сервер (см. раздел 2.4.2).
3. Сконфигурируйте на виртуальной машине `server` первичный DNS-сервер (см. раздел 2.4.3).
4. При помощи утилит `dig` и `host` проанализируйте работу DNS-сервера (см. раздел 2.4.4).
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` (см. раздел 2.4.5).

## 2.4. Последовательность выполнения работы

### 2.4.1. Установка DNS-сервера

1. Загрузите вашу операционную систему и перейдите в рабочий каталог с проектом:

```
cd /var/tmp/user_name/vagrant
```

Здесь `user_name` — идентифицирующее вас имя пользователя, обычно первые буквы инициалов и фамилия.

2. Запустите виртуальную машину `server`:

```
make server-up
```

(или, если вы работаете под ОС Windows, то `vagrant up server`).

3. На виртуальной машине `server` войдите под созданным вами в предыдущей работе пользователем и откройте терминал. Перейдите в режим суперпользователя:

```
sudo -i
```

4. Установите `bind` и `bind-utils`:

```
dnf -y install bind bind-utils
```

5. В качестве упражнения с помощью утилиты `dig` сделайте запрос, например, к DNS-адресу `www.yandex.ru`:

```
dig www.yandex.ru
```

Проанализируйте в отчёте построчно выведенную на экран информацию.

## 2.4.2. Конфигурирование кэширующего DNS-сервера

### 2.4.2.1. Конфигурирование кэширующего DNS-сервера при отсутствии фильтрации DNS-запросов маршрутизаторами

1. В отчёте проанализируйте построчно содержание файлов `/etc/resolv.conf`, `/etc/named.conf`, `/var/named/named.ca`, `/var/named/named.loopback`.
2. Запустите DNS-сервер:  
`systemctl start named`
3. Включите запуск DNS-сервера в автозапуск при загрузке системы:  
`systemctl enable named`
4. Проанализируйте в отчёте отличие в выведенной на экран информации при выполнении команд  
`dig www.yandex.ru`  
и  
`dig @127.0.0.1 www.yandex.ru`
5. Сделайте 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`
6. Сделайте тоже самое для соединения `System eth0` (если оно активно):  
`nmcli connection edit System\ eth0`  
`remove ipv4.dns`  
`set ipv4.ignore-auto-dns yes`  
`set ipv4.dns 127.0.0.1`  
`save`  
`quit`
7. Перезапустите NetworkManager:  
`systemctl restart NetworkManager`  
Проверьте наличие изменений в файле `/etc/resolv.conf`.
8. Требуется настроить направление DNS-запросов от всех узлов внутренней сети, включая запросы от узла `server`, через узел `server`. Для этого внесите 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; };`
9. Внесите изменения в настройки межсетевого экрана узла `server`, разрешив работу с DNS:  
`firewall-cmd --add-service=dns`  
`firewall-cmd --add-service=dns --permanent`
10. Убедитесь, что DNS-запросы идут через узел `server`, который прослушивает порт 53. Для этого на данном этапе используйте команду `lsof`:  
`lsof | grep UDP`

### 2.4.2.2. 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;
```

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

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

Например, для хостов в дисплейном классе мы получим следующие данные для конфигурационного файла `named.conf` виртуальной машины `server`:

```
forwarders { 10.128.0.240; 80.250.174.240; };
forward first;
```

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;
```

### 2.4.3. Конфигурирование первичного DNS-сервера

1. Скопируйте шаблон описания DNS-зон `named.rfc1912.zones` из каталога `/etc` в каталог `/etc/named` и переименуйте его в `user.net` (вместо `user` укажите свой логин):  

```
cp /etc/named.rfc1912.zones /etc/named/
cd /etc/named
mv /etc/named/named.rfc1912.zones /etc/named/user.net
```
2. Включите файл описания зоны `/etc/named/user.net` в конфигурационном файле `DNS /etc/named.conf`, добавив в нём в конце строку:  

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

(вместо `user` укажите свой логин).
3. Откройте файл `/etc/named/user.net` на редактирование и вместо зоны

```
zone "localhost.localdomain" IN {
    type master;
    file "named.localhost";
    allow-update { none; };
};
```

пропишите 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; };
};
```

пропишите 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 удалите.

- В каталоге /var/named создайте 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
```

- Скопируйте шаблон прямой DNS-зоны named.localhost из каталога /var/named в каталог /var/named/master/fz и переименуйте его в user.net (вместо user укажите свой логин):

```
cp /var/named/named.localhost /var/named/master/fz/
cd /var/named/master/fz/
mv named.localhost user.net
```

- Измените файл /var/named/master/fz/user.net, указав необходимые DNS-записи для прямой зоны. В этом файле DNS-имя сервера @ name.invalid. должно быть заменено на @ server.user.net. (вместо user должен быть указан ваш логин); формат серийного номера ГГГГММДДВВ (ГГГГ — год, ММ — месяц, ДД — день, ВВ — номер ревизии) [1]; 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). При этом внимательно отнеситесь к синтаксису в этом файле, а именно к пробелам и табуляции. В результате должен получиться файл следующего содержания:

```
$TTL 1D
@           IN SOA      @ server.user.net. (
                2024072700      ; serial
                1D              ; refresh
                1H              ; retry
                1W              ; expire
                3H )            ; minimum
NS          @
A           192.168.1.1
$ORIGIN user.net.
server      A           192.168.1.1
ns          A           192.168.1.1
```

- Скопируйте шаблон обратной DNS-зоны named.loopback из каталога /var/named в каталог /var/named/master/rz и переименуйте 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
```

- Измените файл /var/named/master/rz/192.168.1, указав необходимые DNS-записи для обратной зоны. В этом файле DNS-имя сервера @ name.invalid. должно быть заменено на @ server.user.net. (вместо user должен быть указан ваш логин); формат серийного номера ГГГГММДДВВ (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 (на данном этапе должна быть задана PTR запись, ставящая в соответствие адресу 192.168.1.1 DNS-адрес ns.user.net). В результате должен получиться файл следующего содержания:

```
$TTL 1D
@           IN SOA      @ server.user.net. (
                2024072700      ; serial
```

```

                1D                ; refresh
                1H                ; retry
                1W                ; expire
                3H )              ; minimum
NS              @
A               192.168.1.1
PTR            server.user.net.
$ORIGIN 1.168.192.in-addr.arpa.
1            PTR            server.user.net.
1            PTR            ns.user.net.

```

9. 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

```

10. В системах с запущенным SELinux все процессы и файлы имеют специальные метки безопасности (так называемый «контекст 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`, введите:

```
getsebool -a | grep named
```

При необходимости дайте `named` разрешение на запись в файлы DNS-зоны:

```

setsebool named_write_master_zones 1
setsebool -P named_write_master_zones 1

```

11. В дополнительном терминале запустите в режиме реального времени расширенный лог системных сообщений, чтобы проверить корректность работы системы:

```
journalctl -x -f
```

и в первом терминале перезапустите DNS-сервер:

```
systemctl restart named
```

Если в логе выдаются сообщения об ошибках, то устраните их и повторно перезапустите DNS-сервер.

## 2.4.4. Анализ работы DNS-сервера

1. При помощи утилиты `dig` получите описание DNS-зоны с сервера `ns.user.net` (вместо `user` должен быть указан ваш логин):

```
dig ns.user.net
```

и проанализируйте его.

2. При помощи утилиты `host` проанализируйте корректность работы DNS-сервера:

```

host -l user.net
host -a user.net
host -t A user.net
host -t PTR 192.168.1.1

```

(вместо `user` должен быть указан ваш логин).

## 2.4.5. Внесение изменений в настройки внутреннего окружения виртуальной машины

1. На виртуальной машине `server` перейдите в каталог для внесения изменений в настройки внутреннего окружения `/vagrant/provision/server/`, создайте в нём каталог `dns`, в который поместите в соответствующие каталоги конфигурационные файлы DNS:

```
cd /vagrant
mkdir -p /vagrant/provision/server/dns/etc/named
mkdir -p /vagrant/provision/server/dns/var/named/master/
cp -R /etc/named.conf /vagrant/provision/server/dns/etc/
cp -R /etc/named/* /vagrant/provision/server/dns/etc/named/
cp -R /var/named/master/*
  └─ /vagrant/provision/server/dns/var/named/master/
```

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

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

Открыв его на редактирование, пропишите в нём следующий скрипт:

```
#!/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
```

Этот скрипт, по сути, повторяет произведённые вами действия по установке и настройке DNS-сервера: подставляет в нужные каталоги подготовленные вами конфигурационные файлы; меняет соответствующим образом права доступа, метки безопасности SELinux и правила межсетевого экрана; настраивает сетевое соединение так, чтобы сервер выступал DNS-сервером по умолчанию для узлов внутренней виртуальной сети; запускает DNS-сервер.



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

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

## 2.5. Содержание отчёта

1. Титульный лист с указанием номера лабораторной работы и ФИО студента.
2. Формулировка задания работы.
3. Описание результатов выполнения задания:
  - скриншоты (снимки экрана), фиксирующие выполнение работы;
  - подробное описание настроек служб в соответствии с заданием;
  - полные тексты конфигурационных файлов настраиваемых в работе служб;
  - результаты проверки корректности настроек служб в соответствии с заданием (подтверждённые скриншотами).
4. Выводы, согласованные с заданием работы.
5. Ответы на контрольные вопросы.

## 2.6. Контрольные вопросы

1. Что такое DNS?
2. Каково назначение кэширующего DNS-сервера?
3. Чем отличается прямая DNS-зона от обратной?
4. В каких каталогах и файлах располагаются настройки DNS-сервера? Кратко охарактеризуйте, за что они отвечают.
5. Что указывается в файле `resolv.conf`?
6. Какие типы записи описания ресурсов есть в DNS и для чего они используются?
7. Для чего используется домен `in-addr.arpa`?
8. Для чего нужен демон `named`?
9. В чём заключаются основные функции `slave`-сервера и `master`-сервера?
10. Какие параметры отвечают за время обновления зоны?
11. Как обеспечить защиту зоны от скачивания и просмотра?
12. Какая запись `RR` применяется при создании почтовых серверов?
13. Как протестировать работу сервера доменных имён?
14. Как запустить, перезапустить или остановить какую-либо службу в системе?
15. Как посмотреть отладочную информацию при запуске какого-либо сервиса или службы?
16. Где храниться отладочная информация по работе системы и служб? Как её посмотреть?
17. Как посмотреть, какие файлы использует в своей работе тот или иной процесс? Приведите несколько примеров.
18. Приведите несколько примеров по изменению сетевого соединения при помощи командного интерфейса `nmcli`.
19. Что такое SELinux?
20. Что такое контекст (метка) SELinux?
21. Как восстановить контекст SELinux после внесения изменений в конфигурационные файлы?
22. Как создать разрешающие правила политики SELinux из файлов журналов, содержащих сообщения о запрете операций?
23. Что такое булевый переключатель в SELinux?
24. Как посмотреть список переключателей SELinux и их состояние?
25. Как изменить значение переключателя SELinux?

При ответах на вопросы рекомендуется ознакомиться с Источниками [1–7].

## Список литературы

1. *Barr D.* Common DNS Operational and Configuration Errors: RFC / RFC Editor. — 02/1996. — DOI: 10.17487/rfc1912.
2. Security-Enhanced Linux. Linux с улучшенной безопасностью: руководство пользователя / М. McAllister, S. Radvan, D. Walsh, D. Grift, E. Paris, J. Morris. — URL: [https://docs-old.fedoraproject.org/ru-RU/Fedora/13/html/Security-Enhanced\\_Linux/index.html](https://docs-old.fedoraproject.org/ru-RU/Fedora/13/html/Security-Enhanced_Linux/index.html) (дата обр. 13.09.2021).
3. Systemd. — 2015. — URL: <https://wiki.archlinux.org/index.php/Systemd> (visited on 09/13/2021).
4. *Костромин В. А.* Утилита lsof — инструмент администратора. — URL: <http://rus-linux.net/kos.php?name=/papers/lsof/lsof.html> (дата обр. 13.09.2021).
5. *Поттеринг Л.* Systemd для администраторов: цикл статей. — 2010. — URL: <http://wiki.opennet.ru/Systemd> (дата обр. 13.09.2021).
6. Сайт проекта NetworkManager. — URL: <https://wiki.gnome.org/Projects/NetworkManager> (visited on 09/13/2021).
7. Сайт проекта nmcli. — URL: <https://developer.gnome.org/NetworkManager/stable/nmcli.html> (visited on 09/13/2021).