

Отчёт по лабораторной работе 2

Настройка DNS-сервера

Чилеше Лупупа

Содержание

1 Цель работы	5
2 Выполнение работы	6
2.1 Первичная проверка DNS-разрешения	6
2.2 Запуск DNS-сервера и анализ результата	7
2.3 Настройка DNS-сервера по умолчанию для интерфейса eth0	8
2.4 Конфигурирование файла named.conf	8
2.5 Разрешение работы DNS-сервера через межсетевой экран	9
2.6 Проверка активности службы DNS	10
2.7 Подключение файла зон	11
2.8 Настройка файла user.net с прямой и обратной зоной	11
2.9 Создание каталогов для файлов зон	12
2.10 Настройка прямой зоны	13
2.11 Настройка обратной зоны	13
2.12 Настройка прав доступа и контекстов SELinux	14
2.13 Перезапуск службы и проверка в логах	14
2.14 Проверка прямой зоны	15
2.15 Проверка зоны домена и обратной зоны	16
2.16 Подготовка каталога provisioning	17
2.17 Создание скрипта dns.sh	17
3 Вывод	19
4 Контрольные вопросы	20

Список иллюстраций

2.1 Результат выполнения dig www.yandex.ru	6
2.2 Сравнение dig и dig [127.0.0.1?]	7
2.3 Настройка NetworkManager	8
2.4 Фрагмент файла named.conf	9
2.5 Результат lsof grep UDP	10
2.6 Файл описания прямой и обратной зон	12
2.7 Файл прямой зоны	13
2.8 Файл обратной зоны	14
2.9 Настройка SELinux и перезапуск named	15
2.10 Проверка A-записи ns.chileshe.net	15
2.11 Проверка прямой и обратной зоны	16
2.12 Создание каталогов и копирование файлов	17
2.13 Содержимое dns.sh	18

Список таблиц

1 Цель работы

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

2 Выполнение работы

2.1 Первичная проверка DNS-разрешения

После установки пакетов bind и bind-utils была выполнена проверка DNS-разрешения для домена `www.yandex.ru` с помощью утилиты `dig`.

На изображении показан результат выполнения команды и содержимое секций HEADER, QUESTION и ANSWER, где отображены три A-записи домена:

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

; <>> DiG 9.18.33 <>> www.yandex.ru
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 13622
;; flags: qr rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 1232
;; QUESTION SECTION:
;www.yandex.ru.           IN      A

;; ANSWER SECTION:
www.yandex.ru.        333     IN      A      77.88.44.55
www.yandex.ru.        333     IN      A      77.88.55.88
www.yandex.ru.        333     IN      A      5.255.255.77

;; Query time: 10 msec
;; SERVER: 10.0.2.3#53(10.0.2.3) (UDP)
;; WHEN: Thu Nov 13 12:23:44 UTC 2025
;; MSG SIZE  rcvd: 90

[root@server.chileshe.net ~]#
```

Рис. 2.1: Результат выполнения `dig www.yandex.ru`

Полученные IP-адреса: - 77.88.44.55

- 77.88.55.88

- 5.255.255.77

Также отображены параметры времени выполнения и адрес DNS-сервера, ответившего на запрос.

2.2 Запуск DNS-сервера и анализ результата

После запуска службы named и включения её в автозагрузку были выполнены два DNS-запроса: обычный и направленный на локальный DNS-сервер.

```
[root@server.chileshe.net ~]# systemctl start named
[root@server.chileshe.net ~]# systemctl enable named
Created symlink '/etc/systemd/system/multi-user.target.wants/named.service' → '/usr/lib/systemd/system/named.service'.
[root@server.chileshe.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: 29487
;; flags: qr rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 1232
;; COOKIE: 4a4cd49aa09729d1010000006915ce0316d0c999d7338023 (good)
;; QUESTION SECTION:
;www.yandex.ru.           IN      A

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

;; Query time: 1601 msec
;; SERVER: 127.0.0.1#53(127.0.0.1) (UDP)
;; WHEN: Thu Nov 13 12:24:35 UTC 2025
;; MSG SIZE rcvd: 118

[root@server.chileshe.net ~]#
```

Рис. 2.2: Сравнение dig и dig [127.0.0.1?]

Отличия:

- Первый запрос использует внешний DNS-сервер.
- Второй – направлен на локальный сервер (127.0.0.1).

Первоначально возникала ошибка связи, так как служба была в процессе настройки. После корректной конфигурации локальный сервер начал выдавать ответы.

2.3 Настройка DNS-сервера по умолчанию для интерфейса eth0

Внутренний DNS-сервер был назначен основным для всех запросов хоста. Для этого параметры соединения eth0 были изменены через NetworkManager.

```
[root@server.chileshe.net ~]#  
[root@server.chileshe.net ~]# nmcli connection edit eth0  
====| nmcli interactive connection editor |====  
Editing existing '802-3-ethernet' connection: 'eth0'  
  
Type 'help' or '?' for available commands.  
Type 'print' to show all the connection properties.  
Type 'describe [<setting>.<prop>]' for detailed property description.  
  
You may edit the following settings: connection, 802-3-ethernet (ethernet), 802-1x, dcb, sriov, ethtool, match, ipv4,  
ipv6, 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  
Connection 'eth0' (e292e83a-7750-4087-b4e1-a998fc55c0ea) successfully updated.  
nmcli> quit  
[root@server.chileshe.net ~]#
```

Рис. 2.3: Настройка NetworkManager

Внесённые изменения: - удалён параметр авто-DNS - отключено автоматическое получение DNS - установлен локальный DNS-сервер: 127.0.0.1

После перезапуска NetworkManager файл /etc/resolv.conf стал использовать локальный адрес.

2.4 Конфигурирование файла named.conf

Для направления DNS-запросов от всех узлов внутренней сети были изменены параметры в /etc/named.conf.

```
named.conf      [-M--] 53 L:[ 1+18 19/ 60] *(662 /1743b) 0125 0x07D
//
// 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.recurse";
    >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.
     *   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;
}
```

Рис. 2.4: Фрагмент файла named.conf

Изменено:

- строка `listen-on port 53 { 127.0.0.1; any; };`
позволяет серверу слушать все интерфейсы.
- строка `allow-query { localhost; 192.168.0.0/16; };`
разрешает DNS-запросы от внутренних сетей.

Эти параметры позволяют сервису named принимать и обрабатывать запросы от всех узлов сети.

2.5 Разрешение работы DNS-сервера через межсетевой экран

Для корректной работы DNS-сервера были разрешены необходимые службы в firewall.

2.6 Проверка активности службы DNS

Для подтверждения того, что named слушает порт 53, был выполнен анализ UDP-портов.

Результаты отображены на изображении.

```
[root@server.chileshe.net ~]# firewall-cmd --add-service=dns
success
[root@server.chileshe.net ~]# firewall-cmd --add-service=dns --permanent
success
[root@server.chileshe.net ~]# 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-dae  894                  avahi    12u    IPv4          8018    0t0    UDP *:mdns
avahi-dae  894                  avahi    13u    IPv6          8019    0t0    UDP *:mdns
chronynd   951                  chrony   5u    IPv4          8650    0t0    UDP localhost:323
chronynd   951                  chrony   6u    IPv6          8651    0t0    UDP localhost:323
named      28273                 named   25u    IPv4          83111   0t0    UDP localhost:domain
named      28273                 named   26u    IPv4          83112   0t0    UDP localhost:domain
named      28273                 named   31u    IPv6          83115   0t0    UDP localhost:domain
named      28273                 named   32u    IPv6          83116   0t0    UDP localhost:domain
named      28273 28274 isc-net-0  named   25u    IPv4          83111   0t0    UDP localhost:domain
named      28273 28274 isc-net-0  named   26u    IPv4          83112   0t0    UDP localhost:domain
named      28273 28274 isc-net-0  named   31u    IPv6          83115   0t0    UDP localhost:domain
named      28273 28274 isc-net-0  named   32u    IPv6          83116   0t0    UDP localhost:domain
named      28273 28275 isc-net-0  named   25u    IPv4          83111   0t0    UDP localhost:domain
named      28273 28275 isc-net-0  named   26u    IPv4          83112   0t0    UDP localhost:domain
named      28273 28275 isc-net-0  named   31u    IPv6          83115   0t0    UDP localhost:domain
named      28273 28275 isc-net-0  named   32u    IPv6          83116   0t0    UDP localhost:domain
named      28273 28276 isc-net-0  named   25u    IPv4          83111   0t0    UDP localhost:domain
```

Рис. 2.5: Результат lsof | grep UDP

На экране видно: - множество открытых UDP-сокетов процесса named - порт domain (53) прослушивается - присутствуют записи для IPv4 и IPv6
Это подтверждает корректную работу кэширующего DNS-сервера.

2.7 Подключение файла зон

Для начала был скопирован шаблон named.rfc1912.zones в каталог /etc/named и переименован согласно варианту. Затем файл был подключён в конфигурации BIND через директиву `include`.

После подключения в конфигурации сервера стала доступна собственная зона, предназначенная для дальнейшего редактирования.

2.8 Настройка файла user.net с прямой и обратной зоной

В файле описания зон были удалены стандартные записи и добавлены две собственные зоны:

- прямая зона домена chileshe.net
- обратная зона 1.168.192.in-addr.arpa

Соответствующий фрагмент файла выглядит следующим образом:

```
chileshe.net      [---] 2 L:[ 2+25 27/ 29] *(698 / 700b) 0010 0x00A
//
// 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 "chileshe.net" IN {
<----->type master;
<----->file "master/fz/chileshe.net";
<----->allow-update { none; };
};

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

Рис. 2.6: Файл описания прямой и обратной зон

2.9 Создание каталогов для файлов зон

В каталоге `/var/named` были созданы подкаталоги:

- `master/fz` – для файлов прямой зоны
- `master/rz` – для файлов обратной зоны

Они предназначены для хранения пользовательских файлов зоны, созданных на основе шаблонов.

2.10 Настройка прямой зоны

Шаблон прямой зоны был скопирован в каталог `master/fz` и переименован.

Далее в него были внесены необходимые изменения:

- заменено имя сервера на `server.chileshe.net`.
- указан корректный серийный номер
- изменён IP-адрес на `192.168.1.1`
- установлена директива `$ORIGIN chileshe.net.`.
- добавлены A-записи для серверов

Конечное содержимое прямой зоны:

```
chileshe.net      [---] 0 L:[ 1+12 13/ 14] *(219 / 220b) 0010 0x00A
$TTL 1D
@<---->IN SOA<>@ server.chileshe.net. (
<----><----><----><----><---->2025111300<---->; serial
<----><----><----><----><---->1D<---->; refresh
<----><----><----><----><---->1H<---->; retry
<----><----><----><----><---->1W<---->; expire
<----><----><----><----><---->3H )<-->; minimum
<---->NS<---->@
<---->A<---->192.168.1.1
$ORIGIN chileshe.net.
server<>A<---->192.168.1.1
ns<>A<---->192.168.1.1
```

Рис. 2.7: Файл прямой зоны

2.11 Настройка обратной зоны

Шаблон обратной зоны был скопирован в каталог `master/rz` и переименован в `192.168.1`. Далее были внесены изменения:

- обновлён SOA-блок

- указан IP-адрес сервера
- добавлены PTR-записи для правильного обратного отображения IP → имя

Готовый файл обратной зоны имеет следующий вид:

```

192.168.1      [---]  0 L:[ 1+13 14/ 15] *(267 / 268b) 0010 0x00A
$TTL 1D
@<---->IN SOA<>@ server.chileshe.net. (
<----><----><----><----><---->2025111300<---->; serial
<----><----><----><----><---->1D<---->; refresh
<----><----><----><----><---->1H<---->; retry
<----><----><----><----><---->1W<---->; expire
<----><----><----><----><---->3H )<--->; minimum
<---->NS<---->@
<---->A<---->192.168.1.1
<---->PTR<---->server.chileshe.net.
$ORIGIN 1.168.192.in-addr.arpa.
1<---->PTR<---->server.chileshe.net.
1<---->PTR<---->ns.chileshe.net.

```

Рис. 2.8: Файл обратной зоны

2.12 Настройка прав доступа и контекстов SELinux

Для корректной работы BIND были исправлены владельцы каталогов `/etc/named` и `/var/named`.

После изменения прав были восстановлены SELinux-контексты, необходимые для работы службы.

Также проверены параметры SELinux, связанные с `named`, и разрешена запись в мастер-зоны при необходимости.

2.13 Перезапуск службы и проверка в логах

После завершения настройки был выполнен перезапуск службы DNS.

В логе (`journalctl -x -f`) проверено отсутствие ошибок. На скриншоте показан фрагмент команд и проверок:

```
[root@server.chileshe.net rz]#
[root@server.chileshe.net rz]# chown -R named:named /etc/named
[root@server.chileshe.net rz]# chown -R named:named /var/named
[root@server.chileshe.net rz]# restorecon -vR /etc
Relabeled /etc/lvm/devices/system.devices from system_u:object_r:lvm_metadata_t:s0 to system_u:object_r:lvm_etc_t:s0
Relabeled /etc/lvm/devices/backup/system.devices-20251113.120720.0005 from system_u:object_r:lvm_metadata_t:s0 to syst
em_u:object_r:lvm_etc_t:s0
Relabeled /etc/NetworkManager/system-connections/eth1.nmconnection from unconfined_u:object_r:user_tmp_t:s0 to unconfi
ned_u:object_r:NetworkManager_etc_rw_t:s0
[root@server.chileshe.net rz]# restorecon -vR /var/named
[root@server.chileshe.net rz]# getsebool -a | grep named
named_tcp_bind_http_port --> off
named_write_master_zones --> on
[root@server.chileshe.net rz]# systemctl restart named
[root@server.chileshe.net rz]#
```

Рис. 2.9: Настройка SELinux и перезапуск named

Система успешно приняла новые файлы зон, а DNS-сервер был перезапущен без ошибок.

2.14 Проверка прямой зоны

После завершения конфигурации прямой зоны было выполнено тестирование с помощью утилиты `dig`.

Запрос к записи `ns.chileshe.net` был успешно обработан локальным DNS-сервером:

```
[root@server.chileshe.net rz]#
[root@server.chileshe.net rz]# dig ns.chileshe.net

; <>> DiG 9.18.33 <>> ns.chileshe.net
;; global options: +cmd
;; Got answer:
;; ->>HEADER<- opcode: QUERY, status: NOERROR, id: 16844
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: 16b0267b722b48c0010000006915d281359442664fffd1fec (good)
;; QUESTION SECTION:
;ns.chileshe.net.           IN      A

;; ANSWER SECTION:
ns.chileshe.net.      86400   IN      A      192.168.1.1

;; Query time: 1 msec
;; SERVER: 127.0.0.1#53(127.0.0.1) (UDP)
;; WHEN: Thu Nov 13 12:43:45 UTC 2025
;; MSG SIZE  rcvd: 88

[root@server.chileshe.net rz]#
```

Рис. 2.10: Проверка A-записи ns.chileshe.net

В ответе указано: - статус NOERROR

- одна А-запись - IP-адрес: 192.168.1.1
- истина: данные получены от локального сервера 127.0.0.1#53

Это подтверждает корректность настройки прямой зоны.

2.15 Проверка зоны домена и обратной зоны

Также были выполнены дополнительные проверки с использованием команды host.

Они подтверждают корректность обработки NS-, SOA-, A- и PTR-записей.

```
[root@server.chileshe.net rz]# host -l chileshe.net
chileshe.net name server chileshe.net.
chileshe.net has address 192.168.1.1
ns.chileshe.net has address 192.168.1.1
server.chileshe.net has address 192.168.1.1
[root@server.chileshe.net rz]# host -a chileshe.net
Trying "chileshe.net"
;; ->HEADER<<- opcode: QUERY, status: NOERROR, id: 29169
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;chileshe.net.           IN      ANY

;; ANSWER SECTION:
chileshe.net.        86400   IN      SOA     chileshe.net. server.chileshe.net. 2025111300 86400 3600 604800 10800
chileshe.net.        86400   IN      NS      chileshe.net.
chileshe.net.        86400   IN      A       192.168.1.1

Received 103 bytes from 127.0.0.1#53 in 1 ms
[root@server.chileshe.net rz]# host -t A chileshe.net
chileshe.net has address 192.168.1.1
[root@server.chileshe.net rz]# host -t PTR 192.168.1.1
1.1.168.192.in-addr.arpa domain name pointer ns.chileshe.net.
1.1.168.192.in-addr.arpa domain name pointer server.chileshe.net.
[root@server.chileshe.net rz]#
```

Рис. 2.11: Проверка прямой и обратной зоны

Полученные результаты:

- `host -l chileshe.net` корректно перечисляет все записи зоны
- `host -t any chileshe.net` отображает:
 - SOA-запись зоны
 - NS-записи

- А-запись сервера
- host -t PTR 192.168.1.1 возвращает название хоста:
 - server.chileshe.net
 - ns.chileshe.net

Обратное разрешение работает корректно.

2.16 Подготовка каталога provisioning

Для автоматизации конфигурации DNS в среде Vagrant были созданы каталоги в дереве /vagrant/provision/server/dns/:

```
[root@server.chileshe.net rz]# 
[root@server.chileshe.net rz]# cd /vagrant/
[root@server.chileshe.net vagrant]# mkdir -p /vagrant/provision/server/dns/etc/named
[root@server.chileshe.net vagrant]# mkdir -p /vagrant/provision/server/dns/var/named/master
[root@server.chileshe.net vagrant]# cp -R /etc/named.conf /vagrant/provision/server/dns/etc/
[root@server.chileshe.net vagrant]# cp -R /etc/named/chileshe.net /vagrant/provision/server/dns/etc/named/
[root@server.chileshe.net vagrant]# cp -R /var/named/master/* /vagrant/provision/server/dns/var/named/master/
[root@server.chileshe.net vagrant]# touch dns.sh
[root@server.chileshe.net vagrant]#
```

Рис. 2.12: Создание каталогов и копирование файлов

В каталоги были скопированы: - файл named.conf - содержимое /etc/named/ - файлы прямой и обратной зоны из /var/named/master/

Эти данные будут использоваться дальше для автоматической конфигурации DNS при подъёме окружения.

2.17 Создание скрипта dns.sh

В каталоге /vagrant/provision/server был создан исполняемый файл dns.sh, содержащий команды:

- установка пакетов BIND

- копирование конфигурационных файлов
- установка необходимых прав
- восстановление SELinux-контекстов
- настройка firewall
- разрешение SELinux-политик
- изменение системного DNS на 127.0.0.1
- перезапуск NetworkManager
- запуск службы named

Готовый файл выглядит следующим образом:

```

1  #!/bin/bash
2  echo "Provisioning script $0"
3  echo "Install needed packages"
4  dnf -y install bind bind-utils
5  echo "Copy configuration files"
6  cp -R /vagrant/provision/server/dns/etc/* /etc
7  cp -R /vagrant/provision/server/dns/var/named/* /var/named
8  chown -R named:named /etc/named
9  chown -R named:named /var/named
10 restorecon -vR /etc
11 restorecon -vR /var/named
12 echo "Configure firewall"
13 firewall-cmd --add-service=dns
14 firewall-cmd --add-service=dns --permanent
15 echo "Tuning SELinux"
16 setsebool named_write_master_zones 1
17 setsebool -P named_write_master_zones 1
18 echo "Change dns server address"
19 nmcli connection edit "eth0" <<EOF
20   remove ipv4.dns
21   set ipv4.ignore-auto-dns yes
22   set ipv4.dns 127.0.0.1
23   save
24   quit
25 EOF
26 systemctl restart NetworkManager
27 echo "Start named service"
28 systemctl enable named
29 systemctl start named
30

```

Рис. 2.13: Содержимое dns.sh

Скрипт полностью автоматизирует настройку DNS-сервера при развёртывании виртуальной машины.

3 Вывод

В ходе работы был развернут и настроен полноценный DNS-сервер на базе BIND, включающий кэширующие и авторитативные функции. Созданы прямые и обратные зоны, настроены файлы конфигурации, исправлены права доступа и восстановлены контексты SELinux. Проведены проверки работы сервера с помощью утилит dig и host, подтверждающие корректность разрешения имён и обратного отображения IP-адресов. Создано автоматизированное окружение provisioning для повторного развёртывания DNS-сервера в среде Vagrant. Все этапы выполнены успешно, что демонстрирует правильную настройку инфраструктуры и работоспособность DNS-системы.

4 Контрольные вопросы

1. Что такое DNS?

DNS – это распределённая система доменных имён, предназначенная для преобразования человекочитаемых доменов (например, example.com) в IP-адреса и обратно.

2. Каково назначение кэширующего DNS-сервера?

Кэширующий DNS-сервер обрабатывает запросы клиентов и сохраняет полученные ответы в кэше, ускоряя последующие обращения к тем же доменам и уменьшая нагрузку на внешние DNS-серверы.

3. Чем отличается прямая DNS-зона от обратной?

Прямая зона сопоставляет доменные имена с IP-адресами, а обратная зона осуществляет обратное преобразование – IP-адресов в доменные имена.

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

Кратко охарактеризуйте, за что они отвечают.

Основные каталоги: - /etc/named.conf – главный конфигурационный файл BIND.

- /etc/named/ – дополнительные конфигурационные файлы зон.
- /var/named/ – файлы прямых и обратных зон.

Эти файлы определяют параметры работы сервера, пути к зонам и правила доступа.

5. Что указывается в файле resolv.conf?

В resolv.conf задаются DNS-серверы, которые будут использоваться системой для разрешения доменных имён.

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

Основные типы: - **A** – привязка домена к IPv4-адресу.

- **AAAA** – привязка к IPv6.

- **NS** – указание серверов зоны.

- **CNAME** – алиас доменного имени.

- **MX** – почтовые серверы домена.

- **PTR** – обратное отображение IP → домен.

- **SOA** – стартовая запись зоны с параметрами обновления.

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

Он используется для организации обратного DNS-разрешения (IP → доменное имя).

8. Для чего нужен демон named?

named – это основной процесс BIND, обеспечивающий обработку DNS-запросов, ведение зон и взаимодействие с другими серверами.

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

- **Master** хранит оригинальные файлы зоны.

- **Slave** получает их по механизму zone transfer и использует для распределения нагрузки и отказоустойчивости.

10. Какие параметры отвечают за время обновления зоны?

Параметры в SOA-записи: - **serial** – версия зоны.

- **refresh** – частота проверки обновлений slave-сервером.

- **retry** – время повторной попытки подключения.

- **expire** – время, после которого данные считаются недействительными.

- **minimum** – время кэширования отрицательных ответов.

11. Как обеспечить защиту зоны от скачивания и просмотра?

Ограничить доступ через директивы allow-transfer и allow-query, разрешив работу только доверенным серверам.

12. Какая запись RR применяется при создании почтовых серверов?

Для почтовых серверов используется запись **MX**.

13. Как протестировать работу сервера доменных имён?

Используются утилиты: - dig

- host

- nslookup

Они позволяют выполнять прямые и обратные DNS-запросы.

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

Через systemd: - systemctl start <service>

- systemctl restart <service>

- systemctl stop <service>

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

Через вывод журнала: - journalctl -xe

- systemctl status <service>

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

Системные журналы находятся в journalctl.

Просмотр: - journalctl -x -f – потоковое отображение лога.

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

Через lsof: - lsof -p <PID>

- lsof | grep named

- lsof -i :53

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

- nmcli connection show – список соединений

- nmcli connection edit eth0 – редактирование соединения

- nmcli connection modify eth0 ipv4.dns 8.8.8.8

- nmcli device up eth0 – активация интерфейса

19. Что такое SELinux?

SELinux – подсистема контроля доступа, обеспечивающая Mandatory Access Control (MAC) в Linux.

20. Что такое контекст (метка) SELinux?

Контекст определяет права доступа процесса или файла и используется системой для принятия решений по безопасности.

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

Команда: - restorecon -vR <path>

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

С помощью утилит: - audit2allow – генерация политик по логам

- audit2why – объяснение причин отказов

23. Что такое булевый переключатель в SELinux?

Это параметр, включающий или отключающий конкретные функции SELinux без изменения политик.

24. Как посмотреть список переключателей SELinux и их состояние?

Командой: - getsebool -a

25. Как изменить значение переключателя SELinux?

Через команду: - setsebool <switch> on|off

Для постоянного изменения: - setsebool -P <switch> on|off