Глава 13

Сетевые службы: DNS, SMTP, iSCSI и NTP

В этой главе рассматриваются четыре системные службы: **система доменных имен (DNS), протокол простой передачи почты (SMTP), интерфейс системы малых компьютеров Интернета (iSCSI) и служба протокола сетевого времени (NTP)**.

Для **DNS** цели RHCE требуют настройки сервера имен только для кэширования. Поэтому в этой книге мы не рассматриваем конфигурацию главного или вторичного DNS-сервера.

Далее мы рассмотрим услуги электронной почты **SMTP**. Linux предлагает ряд альтернативных методов обработки входящей и исходящей электронной почты. RHEL 7 включает в себя две службы SMTP: **sendmail** и **Postfix**. Мы ориентируемся на **Postfix**, агент по умолчанию для пересылки почты **RHEL 7**. **Postfix** был первоначально разработан в конце 1990-х годов как альтернатива sendmail. Он модульный и относительно простой в настройке.

Вы также узнаете, как настроить устройство хранения и постоянное монтирование через протокол **ISCSI**. Устройство хранения известно как цель **iSCSI**, тогда как клиенты известны как инициаторы **iSCSI**.

Наконец, в то время как вы узнали о **NTP**-сервере по умолчанию в **Главе 5**, вы узнаете о конфигурации узлов **NTP** в этой главе.

ВНУТРИ ЭКЗАМЕНА

Domain Name Service.

Изучите цели RHCE, связанные с DNS:

- Настройте сервер имен только для кэширования.
- Устранение проблем DNS-клиента.

Вы узнаете, как устранять проблемы **DNS**-клиента с помощью команд **dig** и **host**.

Служба SMTP.

Задачи, связанные со Службой электронной почты на экзамене RHCE относительно просты:

- Настройка системы для пересылки всей электронной почты на центральный почтовый сервер.

Целью этой задачи является настройка нулевого клиента, то есть системы, которая может пересылать только электронные письма на удаленный сервер. Однако для тестирования в нашей лабораторной среде нам понадобится вторая система, настроенная для приема входящих сообщений электронной почты. Хотя это и не является конкретным требованием **RHCE** (оно касалось целей RHCE для RHEL 6), мы рассмотрим некоторые из более общих настроек агентов передачи почты **(MTA**).

iSCSI Targets and Clients.

В этой главе также рассматривается настройка целей **iSCSI** и клиенты (инициаторы), как описано в следующей задаче:

- Настройка системы в качестве цели **iSCSI** или инициатора, который постоянно монтирует цель **iSCSI**.

Служба сетевого времени

Наконец, одна служба, которая частично покрывалась целями **RHCSA**, основана на **NTP** протоколе. Принимая во внимание, что акцент на RHCSA был сделан на «настройке системы для использования служб времени», цель RHCE предполагает, что вам необходимо более глубокое знание конфигурации серверов NTP:

- Синхронизация времени с использованием других пиров **NTP**.

Кроме того, вам нужно для достижения основных целей RHCE, которые применяются ко всем сетевым услугам, как описано в **главе 11.**

ЦЕЛЬ СЕРТИФИКАЦИИ 13.01

Введение в службы доменных имен

DNS - это служба, которая переводит понятные человеку имена хостов, такие как **www.mheducation.com** к **IP-адресам**, таким как **192.0.2.101, и наоборот**. **DNS** является распределенной базой данных; каждый сервер имеет свою делегированную зону полномочий для одного или нескольких доменов. Служба DNS, связанная с RHEL, - это **Berkeley Internet Name Domain (BIND).** Поскольку ни один отдельный **DNS**-сервер не является достаточно большим, чтобы хранить базу данных для всего Интернета, каждый сервер может передавать запросы на другие DNS-серверы.

RHEL 7 включает в себя другую службу **DNS**, **Unbound**. Пакет **Unbound** не включает в себя все функции **BIND**, но его легко и просто настроить, если вам нужен только безопасный кэширующий преобразователь **DNS**. В этой главе мы рассмотрим настройку как **BIND**, так и **Unbound**. Вы можете использовать любой из них для достижения цели RHCE, связанной с конфигурацией **DNS**.

Сервер имен BIND

Служба **DNS** по умолчанию в **RHEL 7** основана на демоне **named**, включенном в программный пакет **BIND 9.9**, разработанный через консорциум **Internet Systems**. Этот пакет включает команды **rndc**, которую вы можете использовать для управления операциями **DNS**.

Параметры пакета DNS

Чтобы настроить систему как **DNS**-сервер **BIND**, начните с **RPM**, связанных с группой пакетов **DNS-серверов** имен, показанной здесь:

- **bind** Включает базовое программное обеспечение сервера имен и обширную документацию
- **bind-chroot** Добавляет каталоги, которые изолируют BIND в так называемую «chroot-jail», которая ограничивает доступ, если **DNS** скомпрометирован.
- **bind-dyndb-ldap** Предоставляет внутренний подключаемый модуль LDAP для BIND.
- **bind-libs** Добавляет библиотечные файлы, используемые RPM-файлами **bind** и **bind-utils**
- **bind-libs-lite** Включает облегченную версию библиотек **BIND** для клиентских утилит
- **bind-license** Содержит файл лицензии BIND
- **bind-utils** Включает такие инструменты, как **dig** и **host** для запроса DNS-серверов и получения информации об именах хостов и доменах.

К настоящему времени вам должно быть удобно устанавливать эти пакеты с такими командами, как **yum**, из репозиториев программного обеспечения, как описано в главе 7.

!!!!! On the jobs !!!!

RHEL 7 также поддерживает пакет *dnsmasq*, который можно использовать для настройки сервера переадресации DNS с интегрированной службой DHCP в небольшой сети.
!!!!!!!

Различные типы DNS-серверов

Несмотря на то, что доступны дополнительные параметры, существует четыре основных типа **DNS**-серверов:

- **Главный DNS-сервер**, уполномоченный для одного или нескольких доменов, включает записи хоста для этого домена.
- **Подчинённый DNS-сервер**. Вместо **главного DNS-сервера** можно использовать подчиненный DNS-сервер, который использует главный DNS-сервер для данных.
- **DNS-сервер только для кэширования.** Хранит последние запросы, как прокси-сервер. Если он настроен с функциями пересылки, он обращается к другим **DNS**-серверам за запросами, которых нет в текущем кэше.
- **DNS-сервер только для пересылки.** Передает все запросы другим **DNS-серверам**.

Как описано ранее, все, что вам нужно знать для сдачи экзамена RHCE, - это настроить **DNS-сервер только для кэширования**.

Каждый из этих серверов может быть настроен с доступом, ограниченным внутренними сетями или даже только локальной системой. Кроме того, они могут быть настроены как общедоступные **DNS-серверы**, доступные для всего Интернета. Но такой доступ сопряжен с риском, поскольку успешная атака на авторитетный корпоративный **DNS-сервер** может легко скрыть их веб-сайты от веб-браузеров клиентов. Эта атака является формой отказа в обслуживании (**denial of service**).

ЦЕЛЬ СЕРТИФИКАЦИИ 13.02

Минимальные конфигурации DNS-сервера

Вы можете настроить **DNS-серверы** путем непосредственного редактирования соответствующих файлов конфигурации. В этом разделе мы кратко рассмотрим файлы конфигурации, установленные с программными пакетами **BIND и Unbound**. Затем вы узнаете, как настроить сервер имен только для кэширования, а также сервер имен, который включает пересылку на указанные DNS-серверы.

Конфигурационные файлы BIND

Файлы конфигурации **DNS** могут помочь вам настроить систему **Linux** как базу данных **имен хостов и IP-адресов**. Эта база данных может быть кэширована, перечислена в локальной базе данных, или запрос может быть перенаправлены в другую систему. Файлы конфигурации, которые поддерживают использование **DNS в качестве сервера**, описаны в **таблице 13-1**. Хотя таблица содержит ссылки на стандартные файлы баз данных **/var/named**, изменения в таких файлах не требуются для настройки **DNS-сервера** с кэшированием или пересылкой.

Если вы установили пакет **bind-chroot**, дерево каталогов и файлов также будет доступно в каталоге **/var/named/chroot** для запуска **BIND** в изолированной тюрьме **chroot**. Если вы хотите запустить **BIND** в изолированной тюрьме, вам нужно переместить файлы **конфигурации и DNS зоны** в **/var/named/chroot/etc** и **/var/named/chroot/var/named**, а затем разрешите сервис **named-chroot**.

В следующих разделах вы будете экспериментировать с файлом **/etc/named.conf**. Вы должны поддержать это некоторым способом. Просто помните о владельце и, да, о контекстах **SELinux** файла, как показано в этих выходных данных:

```
# ls -Z /etc/named.conf
-rw-r-----. root named system_u:object_r:named_conf_t:s0 /etc/named.conf
```

Если резервное копирование восстанавливается случайным образом даже пользователем **root**, владение группой и/или **контексты SELinux** могут быть потеряны. Поэтому, если когда-либо произойдет сбой в запуске или перезапуске именованной службы, проверьте **владельца** и **контекст SELinux файла /etc/named.conf**. При необходимости примените к этому файлу следующие команды:

 # chgrp named /etc/named.conf
 # restorecon -F /etc/named.conf

Кроме того, после тестирования конфигурации **DNS** некоторая информация может остаться в кэше. Такова природа кэширующего **DNS-сервера**. Если этот кэш все еще существует после изменения **DNS** файлы конфигурации, это может повлиять на результаты. Поэтому, разумно очищать кэш DNS после каждого изменения конфигурации с помощью следующей команды:

 # rndc flush

ТАБЛИЦА 13-1 Файлы конфигурации DNS-сервера

Файл конфигурации DNS	Описание
/etc/sysconfig/named	Определяет параметры, которые будут переданы именованному демону при запуске.
/etc/named.conf	Основной файл конфигурации **DNS**. Включает расположение файлов зоны. Может включать данные из других файлов, обычно в Каталог **/etc/named** с директивой **include**.
/etc/named.rfc1912.zones	Добавляет соответствующие зоны для имен и адресов локальных хостов.
/var/named/named.empty	Включает файл зоны шаблона.
/var/named/named.localhost	Перечисляет файл зоны для локального компьютера.
/var/named/named.loopback	Перечисляет файл зоны для адреса обратной связи.

Сервер имен BIND только для кэширования

Когда вы запрашиваете веб-страницу, такую как www.mcgraw-hill.com, запрос на разрешение имени хоста отправляется на настроенный **DNS-сервер**. Ответ - связанный IP-адрес. Запрос также известен как **запрос имени**. Для запросов к **внешним DNS-серверам** ответы могут занимать время. Вот где может помочь сервер имен только для кэширования, поскольку повторяющиеся запросы хранятся локально.

При настройке сервера имен только для кэширования первым шагом является поиск версии по умолчанию **файла конфигурации /etc/named.conf**. Директивы в версии этого файла по умолчанию организованы для настройки сервера имен только для кэширования. Один вид этого файла показан на **рисунке 13-1**.

!!!!!! EXAM !!!!!
В версии по умолчанию *etc/named.conf* настроен для сервера имен только для кэширования, ограниченного системой localhost. Незначительные изменения необходимы, чтобы открыть этот сервер для локальной сети.
!!!!!!

- Директива **options** охватывает несколько основных директив **DNS**, включая следующие:
  - Директивы **listen-on port (и listen-on-v6 port)** определяют номер порта для прослушивания (для **IPv4 и IPv6**).

Чтобы распространить это на локальную сеть, вам необходимо указать IP-адрес локального сетевого интерфейса. Например, если вы хотите, чтобы сервер отвечал на запросы по локальному **IPv4-адресу 192.168.122.50**, измените директиву следующим образом (не забывайте точку с запятой, за которой следует пробел после каждого IP-адреса):

listen-on port 53 { 127.0.0.1; 192.168.122.50; };

Если в локальной сети **активна сеть IPv6**, вам потребуется настроить аналогичные адреса IPv6 для директивы **listen-on-v6**. Если сеть IPv6 не активна, достаточно директивы **listen-on-v6** по умолчанию.

- Директива **directory** указывает, где **DNS-сервер** ищет файлы данных. Помните, что если RPM-пакет **bind-chroot** установлен, эти пути к файлам относятся к **/var/named/chroot**.
- Директива **dump-file** указывает файл, в который **BIND** создает дамп кеша для текущей базы данных **DNS** при выполнении команды **rndc dumpdb**.

РИСУНОК 13-1 Файл конфигурации /etc/named.conf для сервера имен только для кэширования

```
options {
        listen-on port 53 { 127.0.0.1; };
        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";
        allow-query     { localhost; };

        /*
         - 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 yes;
        dnssec-validation yes;
        dnssec-lookaside auto;

        /* Path to ISC DLV key */
        bindkeys-file "/etc/named.iscdlv.key";

        managed-keys-directory "/var/named/dynamic";

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

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";
```

- Директива **statistics-file** указывает файл для записи статистических данных при выполнении команды **rndc stats.**
- Директива **memstatistics-file** указывает место для сохранения статистики использования памяти при выходе из **BIND**.
- Директива **allow-query** перечисляет **IP-адреса**, которым разрешено получать информацию с этого сервера. По умолчанию доступ ограничен локальной системой. Чтобы распространить это на другую сеть, такую как **192.168.122.0/24**, вы должны изменить директиву на это:

  allow-query {127.0.0.1; 192.168.122.0/24; };

- Директива **recursion** включает рекурсивные запросы. Рекурсивный запрос опросит официальные серверы имен для запрошенного домена и всегда предоставит ответ клиентам. Это поведение, которое вы ожидаете от кэширующего сервера имен. Как показано в комментариях к файлу **named.conf** на **рис. 13-1**, если сервер имеет публичный IP-адрес, вы должны ограничить доступ легитимным клиентам с помощью директивы **allow-query**.
- Начиная с версии 9.5 BIND в программное обеспечение включена поддержка расширения безопасности **DNS (DNSSEC)** с директивами **dnssec- *. DNSSEC** защищает сервер имен кэширования от атак спуфинга и отравления кэша, проверяя целостность и подлинность ответов, полученных от других серверов имен. Следующие директивы включают защиту **DNSSEC**, проверку (для проверки подлинности) и выполнение запросов с помощью указанного **bindkeys-file:**

dnssec-enable yes;
dnssec-validation yes;
dnssec-lookaside auto;
bindkeys-file "/etc/named.iscdlv.key";
managed-keys-directory "/var/named/dynamic";

- Директива **logging** определяет еще несколько параметров; директива **channel** определяет методы вывода, в данном случае **default_debug**, активированные в файле **named.run** в каталоге **/var/named/data**, регистрируя только динамические проблемы.
- Директива **zone «.»** Определяет корневую зону для Интернета вместе с корневыми **DNS**-серверами, как указано в файле **/var/named/named.ca**.
- Наконец, директивы **include** включают настройки **localhost**, описанные в файле **/etc/named.rfc1912.zones** вместе с ключом для протокола безопасности **DNSSEC**, хранящимся в файле **/etc/named.root.key**.

Для создания кэширующего DNS-сервера никаких изменений не требуется. Все, что вам нужно сделать, это установить вышеупомянутые пакеты **bind- *** и запустить указанную службу с помощью следующей команды:

 # systemctl start named

Далее выполните команду **rndc status**. В случае успеха вы увидите вывод, аналогичный показанному на **рисунке 13-2. Команда rndc** - это утилита управления сервером имен.

РИСУНОК 13-2 Статус работающего DNS-сервера

```
[root@server1 ~]# rndc status
version: 9.9.4-RedHat-9.9.4-14.el7_0.1 <id:8f9657aa>
CPUs found: 1
worker threads: 1
UDP listeners per interface: 1
number of zones: 101
debug level: 0
xfers running: 0
xfers deferred: 0
soa queries in progress: 0
query logging is OFF
recursive clients: 0/0/1000
tcp clients: 0/100
server is up and running
[root@server1 ~]#
```

Старт named

После запуска **DNS-сервера** командой **systemctl start named**, просмотрите журнал **systemd** с помощью команды **journalctl -u named**. Если есть проблемы, вы увидите сообщения об ошибках. В журнале обычно отображается файл с ошибками. Затем вы можете остановить службу с помощью команды **rndc stop** или **systemctl stop named**, а затем проверить соответствующие файлы конфигурации.

Как только вы будете удовлетворены новой конфигурацией, убедитесь, что **DNS** запускается при следующей перезагрузке **Linux**. Как отмечалось в других главах, следующая команда гарантирует, что демон **named** запускается при следующей загрузке Linux с установками по умолчанию:

 # systemctl enable named

Сервер переадресации имен

Этот тип **DNS-сервера** прост. Требуется одна строка конфигурации в файле **/etc/named.conf**. Как видите, это просто; мы настроили его на пару других DNS-серверов в нашей сети:

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

```
        listen-on-v6 port 53 { ::1; };
        directory "/var/named";
        forward only;
        forwarders {
                192.168.122.1;
                192.168.0.1;
        };
};
```

В этой конфигурации запросы к локальному серверу имен пересылаются на **DNS**-серверы с **указанными IP-адресами**. В домашней лаборатории обычно это серверы имен вашего интернет-провайдера.

Если вы хотите открыть этот сервер для внешних запросов, потребуется еще пара изменений. Изменения такие же, как и ранее, в конфигурации сервера имен только для кэширования. Например, если локальная сетевая карта имеет адрес 192.168.122.50, вы бы изменили директиву **listen-on port**

listen-on port 53 { 127.0.0.1; 192.168.122.50; };

Вы также должны включить директиву **allow-query**, описанную ранее, со ссылками на систему **localhost** и **адрес локальной сети**:

allow-query { localhost; 192.168.122.0/24; };

Не забудьте включить **службу DNS** на локальном брандмауэре:

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

Пересылка с сервера имен только для кэширования

Как указывалось ранее, **caching-only name server**, настроенный только для кэширования и настроенный в версии файла **/etc/named.conf** по умолчанию, включен для рекурсивных запросов. В противном случае он не сможет вернуть результаты **DNS-запросов** для зон, для которых он не является доверенным сервером.

Однако вы можете комбинировать аспекты только что описанных серверов кэширования и пересылки имен. Запросы, отсутствующие в локальном кэше, будут перенаправляться на серверы имен, указанные в директиве пересылки. На **рисунке 13-3** показан соответствующий фрагмент файла **/etc/named.conf**, в который были включены директивы пересылки.

рисунок 13-3 Кэширующий сервер имен, который перенаправляет на определенные DNS-серверы.

```
//
// 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; 192.168.122.50; };
        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";
        allow-query     { localhost; };

        forwarders {
                192.168.122.1;
                192.168.0.1;
        };
```

Команды BIND

Две полезные команды, связанные со **службой BIND**, это **named-checkconf** и **rndc**. Команда **named-checkconf** проверяет файл **/etc/named.conf** на наличие синтаксических ошибок. Если ошибок не найдено, он выходит со **статусом 0**; в противном случае на экране отображаются проблемные строки конфигурации.

Аргументы команды **rndc** просты. Попробуйте **rndc** сам по себе. Вывод проведет вас через доступные параметры. Опции, которые мы используем, просты: **ndc status, rndc flush, rndc reload**, и **rndc stop**. Если **DNS-сервер** работает правильно, команда **rndc status** должна отобразить результаты, показанные на **рисунке 13-2**. Команда **rndc flush** очищает кэш сервера. Команда **rndc reload** перечитывает любые изменения, внесенные в файлы конфигурации или зоны DNS. Наконец, команда **rndc stop** останавливает работу **DNS-сервера**.

Unbound как кеширующий сервер имен

Если вам не нужен полнофункциональный сервер имен, такой как **BIND**, вы можете выбрать **Unbound DNS resolver**. Это небольшой пакет, который обеспечивает кеширование и пересылку имен. Проект **Unbound** изначально финансировался VeriSign. Программное обеспечение в настоящее время поддерживается NLnet Labs и распространяется под лицензией BSD. Он был разработан с учетом требований безопасности и модульности и поэтому является жизнеспособной альтернативой **BIND** в качестве локального преобразователя **DNS**.

Чтобы установить **Unbound**, выполните следующую команду:

 # yum install unbound

Файл конфигурации по умолчанию - **/etc/unbound/unbound.conf**. Хотя файл содержит более 500 строк, он содержит множество комментариев и примеров. Команда **man unbound.conf** предоставляет дополнительную информацию и некоторые примеры конфигурации.

!!!!! EXAM !!!!!
Экзамены Red Hat основаны на лабораторных условиях, поэтому результаты имеют значение, а не то, как вы их достигнете. Поэтому, если лабораторный вопрос не требует от вас установки *BIND* или *Unbound*, не стесняйтесь выбирать любой из них для настройки сервера имен кэширования.
!!!!!!!!

Чтобы настроить **сервер имен кэширования/пересылки**, вам нужно включить только три директивы в файле **unbound.conf**. Во-первых, вы должны указать, какие интерфейсы **Unbound** должны прослушивать:

interface: 0.0.0.0

Если вы не включили директиву **interface** в файл конфигурации, **Unbound** прослушивает только интерфейс **localhost**. Директива **interface** аналогична параметрам **listen-on port** и **listen-on-v6 port** в **BIND**. Вы можете указать **IP-адрес** локального интерфейса или 0.0.0.0 для привязки ко всем интерфейсам **IPv4**. Если **Unbound** прослушивает интерфейс, отличный от **localhost**, включите службу **DNS на локальном брандмауэре**.

Далее укажите, каким клиентам разрешено отправлять запросы на сервер:

access-control: 192.168.122.0/24 allow

Директива **access-control** имеет такую же функцию, как **allow-query** в **BIND**. Файл **unbound.conf** содержит несколько закомментированных примеров допустимых строк конфигурации:

 # access-control: 0.0.0.0/0 refuse
 # access-control: 127.0.0.0/8 allow
 # access-control: ::0/0 refuse
 # access-control: ::1 allow

 # access-control: ::ffff:127.0.0.1 allow

Без директивы **access-control Unbound** разрешает клиентские запросы только с локального хоста.

При необходимости настройте пересылку для отправки **DNS-запросов** на другой сервер имен. Как и в конфигурации **access-control**, вам нужно определить **зону с именем «.»** Для пересылки всех запросов на сервер имен:

forward-zone:
name: "."
forward-addr: 192.168.0.1

Наконец, проверьте синтаксис конфигурации с помощью команды **unbound-checkconf**. Запустите и включите сервис **unbound** с помощью команд, перечисленных далее:

 # systemctl start unbound
 # systemctl enable unbound

Устранение неполадок DNS-клиента

После настройки преобразователя **DNS** проверьте результаты с помощью команды, такой как команда **host mheducation.com localhost**. Вывод подтверждает использование локальной системы в качестве **DNS-сервера**, а затем обеспечивает прямое представление **IP-адреса** хоста и имени хоста почтового сервера:

Using domain server:
Name: localhost
Address: 127.0.0.1#53
Aliases:
mheducation.com has address 204.74.99.100
mheducation.com mail is handled by 20 mheducation-com.mail.protection.outlook.com

Вы можете использовать команду **dig** или **host** для проверки вашей работы. Например, с помощью команды **dig @127.0.0.1 www.mheducation.com** вы увидите что-то вроде вывода, показанного на **рисунке 13-4**.

РИСУНОК 13-4 Протестируйте локальный DNS-сервер с помощью команды dig.

```
; <<>> DiG 9.9.4-RedHat-9.9.4-14.el7_0.1 <<>> @127.0.0.1 www.mheducation.com
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53296
;; flags: qr rd ra; QUERY: 1, ANSWER: 4, AUTHORITY: 4, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;www.mheducation.com.           IN      A

;; ANSWER SECTION:
www.mheducation.com.    600     IN      CNAME   ecom-prod-ext-460002190.us-east
1.elb.amazonaws.com.
ecom-prod-ext-460002190.us-east-1.elb.amazonaws.com. 60 IN A 52.1.15.205
ecom-prod-ext-460002190.us-east-1.elb.amazonaws.com. 60 IN A 54.175.172.124
ecom-prod-ext-460002190.us-east-1.elb.amazonaws.com. 60 IN A 52.0.232.222

;; AUTHORITY SECTION:
us-east-1.elb.amazonaws.com. 299 IN     NS      ns-1119.awsdns-11.org.
us-east-1.elb.amazonaws.com. 299 IN     NS      ns-934.awsdns-52.net.
us-east-1.elb.amazonaws.com. 299 IN     NS      ns-235.awsdns-29.com.
us-east-1.elb.amazonaws.com. 299 IN     NS      ns-1793.awsdns-32.co.uk.

;; Query time: 4901 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Mon Nov 30 00:25:17 GMT 2015
;; MSG SIZE  rcvd: 295
```

Команда **dig**, показанная на рисунке, просит локальный **DNS-сервер** найти «**А запись**» на **www.mheducation.com**. **Запись А** отображает **имя хоста в IP-адрес**. Предполагая, что информация об **IP-адресе для www.mheducation.com** не кэшируется локально, затем он связывается с одной из прямых **DNS-систем**, перечисленных в файле **named.conf**. Если эти системы не работают или иным образом недоступны, **локальный DNS-сервер пересылает** запрос на один из серверов имен, указанных в файле **named.ca**. Как будто это корневые серверы имен для Интернета, запрос будет передан другому **DNS-серверу**, который является **полномочным (authoritative)** для домена **mheducation.com**. Поэтому может пройти несколько секунд, прежде чем вы увидите ответ.

В разделе ответов, показанном на **рисунке 13-4**, похоже, что **www.mheducation.com** на самом деле является **псевдонимом (CNAME),** который указывает на другое имя хоста. Команда **dig** может запрашивать все типы записей ресурсов **DNS** с помощью **ключа -t**. Например, чтобы определить почтовые серверы для домена **mheducation.com**, запросите запись **MX (mail exchange)** с помощью следующей команды:

```
# dig -t MX mheducation.com
```

Как вы заметили, существуют разные типы записей ресурсов **DNS**. Наиболее распространенные из них приведены в **таблице 13-2**.

ТАБЛИЦА 13-2 Наиболее распространенные записи ресурсов DNS

Запись ресурса DNS	Описание
A	Сопоставляет имя хоста с **IPv4-адресом**
AAAA	Сопоставляет имя хоста с **IPv6-адресом**
PTR	Сопоставляет **IP-адрес с именем хоста**
CNAME	Псевдоним; сопоставляет имя хоста с другим именем хоста
NS	Возвращает серверы имен, которые являются **полномочными(authoritative)** для **зоны DNS**
MX	Возвращает **почтовые серверы для зоны DNS**
SOA	Возвращает информацию о **зоне DNS**

УПРАЖНЕНИЕ 13-1

Настройте свой собственный DNS-сервер BIND

Следуя приведенным выше файлам примеров, настройте локальный кэширующий **DNS-сервер**, используя сервер имен **BIND**. Доступ будет ограничен локальной системой.

1. Установите **RPM-пакет bind**.
2. Просмотрите содержимое файла **/etc/named.conf**, основываясь на обсуждении, которое было проведено в этой главе. Не вносите никаких изменений.
3. Запустите **DNS-сервер** с помощью следующей команды:

```
# systemctl start named
```

4. Чтобы убедиться, что **DNS-сервер** работает, выполните команду **rndc status**. Вывод должен быть аналогичен показанному на **рисунке 13-2**. Сравните вывод с выводом команды **systemctl status named**.
5. Очистите текущий кэш с помощью команды **rndc flush**.
6. Проверьте **DNS-сервер**. Попробуйте команду **dig @127.0.0.1 www.mheducation.com.**
7. Остановите службу **BIND** с помощью команды **systemctl stop named**

УПРАЖНЕНИЕ 13-2

Настройте свой собственный Unbound DNS-сервер

Требования для этого упражнения идентичны предыдущему. Однако вы будете использовать сервер имен **Unbound**, а не **BIND**.

1. Установите RPM пакет **Unbound**.
2. Просмотрите файл конфигурации **/etc/unbound/unbound.conf**. Не вносите никаких изменений.
3. Запустите DNS-сервер с помощью следующей команды:

   # systemctl start unbound

4. Проверьте **DNS-сервер**. Попробуйте команду **dig @127.0.0.1 www.mheducation.com.**

ЦЕЛЬ СЕРТИФИКАЦИИ 13.03

Различные агенты электронной почты

Конфигурационные файлы **Postfix** могут показаться многословными для инженеров **Linux**, новичков в администрировании электронной почты. Не позволяйте размеру файлов конфигурации запугивать вас. Требуется всего несколько изменений, чтобы удовлетворить требования, связанные с целью RHCE. В этом разделе вы узнаете, где службы SMTP вписываются в иерархию служб электронной почты.

ТАБЛИЦА 13-3 Компоненты почтового сервера

Сокращение	Значение	Примеры
MTA	Агент пересылки почты	**Postfix, sendmail, Dovecot**
MUA	Почтовый пользовательский агент	**mutt, Evolution, mail, Thunderbird**
MDA	Агент доставки почты	**procmail**
MSA	Агент отправки почты	**Postfix, sendmail**

Определения и протоколы

Почтовый сервер состоит из четырех основных компонентов, как описано в **таблице 13-3**. На любом компьютере с **Linux** вы можете настроить агент передачи почты (**MTA**), такой как **Postfix** или **sendmail**, для различных исходящих сервисов, таких как пересылка, ретрансляция, обмен данными между промежуточным узлом и другими **MTA**, псевдонимами и каталогами спулинга. Другие **MTA**, такие как **Dovecot**, предназначены для обработки только входящих служб электронной почты на основе протоколов, которые они обслуживают, **POP3** (**Post Office Protocol, version 3**) и **IMAP4** (**Internet Message Access Protocol, version 4**).

Системы электронной почты сильно зависят от правильного разрешения имен. Хотя вы можете обрабатывать разрешение имен через **/etc/hosts** в небольшой сети, любой почтовой системе, подключенной к Интернету, необходим доступ к полнофункциональному **DNS-серверу**. Для защиты от спама и многого другого важно убедиться, что система, которая намеревается отправить электронное письмо, имеет действительную обратную запись **DNS (PTR)** и действительно передает с **этим IP-адресом.**

Но это только один компонент работы электронной почты, от передачи до доставки. Сообщения электронной почты начинаются с почтового агента пользователя (**MUA**), клиентской системы для отправки и получения электронной почты, такой как **Mutt, Evolution** или **Thunderbird**. С помощью агента отправки почты (**MSA**) такая почта обычно отправляется в **MTA**, такой как **Postfix** или **sendmail**. Агент доставки почты (**MDA**), такой как **procmail**, работает локально для передачи электронной почты с сервера в папку входящих сообщений. **procmail** также может быть использован для фильтрации электронной почты. **Red Hat** поддерживает дополнительные сервисы **MTA**, такие как **Dovecot**, чтобы **POP3** и/или **IMAP** (или защищенные родственники, **POP3** и **IMAP**) могли получать электронную почту.

SMTP, простой протокол передачи почты (**Simple Mail Transfer Protocol**), стал одним из важнейших сервисных протоколов современной эпохи. Большая часть мира, подключенного к Интернету, живет и умирает по электронной почте и полагается на **SMTP** для ее доставки. Как и **POP3** и **IMAP**, **SMTP** - это протокол, набор правил для передачи данных, используемых различными агентами пересылки почты.

Соответствующие пакеты почтового сервера

Пакеты, связанные с **Postfix**, являются частью группы пакетов «**E-mail Server**». Ключевые пакеты перечислены в **таблице 13-4**. Вы можете установить их с помощью команды **rpm** или **yum**. Просто помните, что вам не нужно устанавливать все из этой таблицы.

ТАБЛИЦА 13-4 Пакеты Почтового Сервера

RPM пакет	Описание
cyrus-imapd-*	Устанавливает корпоративную систему электронной почты **Cyrus IMAP.**
cyrus-sasl	Добавляет реализацию **Cyrus** уровня простой аутентификации и безопасности (**SASL**).
dovecot	Поддерживает протоколы входящей электронной почты **IMAP** и **POP**.
dovecot-mysql, dovecot-pgsql, dovecot-pigeonhole	Включает в себя базу данных и соответствующие плагины для **Dovecot**.
mailman	Поддерживает списки обсуждений по электронной почте.
postfix	Почтовый сервер по умолчанию на **RHEL 7**. Это альтернатива **sendmail**.
sendmail	Устанавливает самый популярный почтовый сервер с открытым исходным кодом с тем же именем.
sendmail-cf	Добавляет несколько шаблонов, которые вы можете использовать для генерации вашего файла конфигурации **sendmail**; требуется обработать несколько файлов конфигурации **sendmail**.
spamassassin	Включает в себя пакет фильтра спама с тем же именем.

После установки в группу пакетов почтового сервера по умолчанию входят пакеты программного обеспечения для серверов **Postfix** и **Dovecot**, а также фильтр **SpamAssassin**. Для проведения экзамена **RHCE** вам не нужны все эти пакеты, только **Postfix**. Установите **Postfix** с помощью команды **rpm** или **yum**, если он не установлен по умолчанию.

Используйте альтернативную команду для выбора системы электронной почты

Команда **alternatives**, с параметром **--config**, поддерживает выбор между различными сервисами, такими как **Postfix** и **sendmail**:

 # alternatives --config mta

Команда приводит к следующему выводу, который позволяет выбрать один из установленных **SMTP-серверов** электронной почты. Другие службы SMTP, если они установлены, будут включены в следующий список:

There are 2 programs which provide 'mta'.

Selection Command

***+ 1 /usr/sbin/sendmail.postfix**
2 /usr/sbin/sendmail.sendmail

Enter to keep the current selection[+], or type selection number:

В предыдущем выводе предполагается, что **Postfix** и **sendmail** установлены в системе.

Команда **alternatives** не останавливает и не запускает службу. Если вы не остановите исходную почтовую службу, демон все еще будет работать. Важно, чтобы в системе работала только одна служба **SMTP**. Взаимодействие между **sendmail** и **Postfix** может привести к ошибкам.

В этой главе мы предполагаем, что вы используете почтовый агент **Postfix**. Вы можете подтвердить, что **Postfix** является **MTA** по умолчанию с помощью следующей команды:

```
# alternatives --list | grep mta
mta auto /usr/sbin/sendmail.postfix
```

Общая безопасность пользователя

По умолчанию всем пользователям разрешено использовать локально настроенные службы **SMTP** без использования паролей. Вы увидите, как это можно изменить для **Postfix**, позже в этой главе.

В некоторых случаях вы можете настроить локальных пользователей только для того, чтобы они имели доступ к таким службам. Если вы не хотите, чтобы такие пользователи входили на сервер с обычными учетными записями, один из вариантов - убедиться, что у таких пользователей нет оболочки входа. Например, следующая команда может настроить пользователя с именем **tempworker** в локальной системе без оболочки входа в систему:

```
# useradd tempworker -s /sbin/nologin
```

Пользователь **tempworker** может затем настроить свой собственный менеджер электронной почты, такой как **Evolution**, **Thunderbird** или даже **Outlook Express**, для подключения к сетевым службам **Postfix** или **sendmail SMTP**. Любые попытки этого пользователя открыть **сеанс SSH** на сервере отклоняются.

Конечно, доступ ограничен сконфигурированными пользователями, независимо от того, настроены ли их учетные записи с помощью оболочки входа в систему. Это настроено благодаря простой аутентификации и уровню безопасности (**Simple Authentication and Security Layer SASL**). Как реализовано в **RHEL 7**, это основано на пакете **cyrus-sasl**, настроенном в каталоге **/etc/sasl2**. Файл конфигурации для **Postfix (smtpd.conf)** ссылается на ту же схему аутентификации со следующей директивой:

```
pwcheck_method:saslauthd
```

Файл конфигурации **/etc/sysconfig/saslauthd** подтверждает стандартный механизм проверки паролей следующей директивой:

```
MECH=pam
```

Это ссылка на сменные модули аутентификации (**Pluggable Authentication Modules PAM**), описанные **в главе 10**. Другими словами, пользователи, которые настроены в локальной системе, управляются связанным файлом в каталоге **/etc/pam.d**, а именно **smtp.postfix** и **smtp.sendmail** для **Postfix** и **sendmail** соответственно. Тем не менее, вам нужно внести несколько изменений в **Postfix**, чтобы он действительно читал базу данных аутентификации.

Логирование почты

Большинство сообщений журнала, связанных со службами **SMTP**, можно найти в файле **/var/log/maillog**. Сообщения, которые вы можете ожидать увидеть в этом файле, относятся к

- Restarts(перезагрузка) of Postfix
- Successful and failed user connections
- Sent and rejected(отклонённые) e-mail messages

Общие проблемы безопасности

По умолчанию служба **SMTP** использует **порт 25**. Если вы откроете **порт 25 на брандмауэре**, внешние пользователи могут иметь доступ к этому серверу. Вы можете открыть этот порт в зоне по умолчанию с помощью следующих команд:

```
# firewall-cmd --permanent --add-service=smtp
# firewall-cmd --reload
```

Чтобы создать **настраиваемое правило**, которое поддерживает доступ только из систем в сети **192.168.122.0/24**, вы можете добавить расширенное правило с помощью следующей команды:

 # firewall-cmd --permanent --add-rich-rule='rule family=ipv4 source address=192.168.122.0/24 service \ name=smtp accept'

В общем, **SELinux** не является проблемой для служб **SMTP**. Только один логический **SELinux** применяется к службе **Postfix**, **allow_postfix_local_write_mail_spool**. Он активен по умолчанию. Как следует из названия, он позволяет службе **Postfix** записывать файлы электронной почты в локальный области каталога **/var/spool postfix**.

Тестирование почтового сервера

Помимо команды **telnet**, описанной далее в этой главе, подходящим способом тестирования почтового сервера является почтовый клиент. Конечно, было бы удобно иметь почтовый клиент с графическим интерфейсом; однако, как обсуждалось в **главе 2**, могут быть доступны только текстовые клиенты, такие как **mutt**.

УПРАЖНЕНИЕ 13-3
Создать пользователей только для электронной почты

В этом упражнении вы создадите трех пользователей в локальной системе, чтобы они могли обращаться к локальному **SMTP-серверу**. Понятно, что для настройки доступа или ограничений для этих пользователей на **SMTP-сервере Postfix** требуется дополнительная конфигурация. Пользователями являются **mailer1, mailer2** и **mailer3**.

1. Просмотрите команду **useradd**. Определите переключатель, связанный с оболочкой входа по умолчанию.
2. Просмотрите содержимое файла **/etc/passwd**. Найдите оболочку, которая не позволяет входить в систему:

   /sbin/nologin

3. Запустите такие команды, как **useradd mailer1 -s /sbin/nologin**, чтобы добавить нового пользователя. Обязательно назначьте этому пользователю пароль.
4. Просмотрите результат в **/etc/shadow**.
5. Повторите шаг 3 для пользователей **mailer2** и **mailer3**.
6. Попробуйте войти в одну из новых учетных записей как обычный пользователь. Это должно потерпеть неудачу. Просмотрите связанные сообщения в файле **/var/log/secure**.
7. Сохраните новых пользователей.

ЦЕЛЬ СЕРТИФИКАЦИИ 13.04

Конфигурация Postfix

Почтовый сервер **Postfix** - это один из способов управления потоком электронной почты в системе и сети. Стандартные файлы конфигурации хранятся в каталоге **/etc/postfix**. Команду **postconf** можно использовать для проверки конфигурации. Как установлено, **Postfix** принимает электронную почту только из локальной системы. Изменения конфигурации, необходимые для настройки **Postfix** на прием входящей электронной почты и пересылку электронной почты через промежуточный узел, относительно просты.

Для целей этой главы **Postfix** был установлен на систему физического хоста. Другой сервер **Postfix** был установлен на **server1.example.com** и настроен для пересылки электронной почты на центральный почтовый сервер, работающий на физическом хосте. Тесты доступа были выполнены с виртуальных машин, настроенных в **главах 1 и 2**, представляющих различные внешние сети.

Подробная информация о файлах конфигурации **Postfix** включает опции для безопасности на уровне пользователя и хоста. Если вы уже знаете, как настроить **Postfix** для основной операции и просто

хотите знать, что требуется для достижения целей **SMTP** для **Postfix**, перейдите к разделу, связанному с настройкой **Postfix** в качестве нулевого клиента.

Конфигурационные файлы

Файлы конфигурации хранятся в каталоге **/etc/postfix**. Основной файл конфигурации, **main.cf**, несколько проще, чем альтернатива **sendmail**, **sendmail.cf**. Это все еще сложно, так как включает в себя почти 700 строк.

За исключением файлов **.cf**, любые изменения должны сначала обрабатываться в базе данных с помощью команды **postmap**. Например, если вы добавили ограничения к файлу доступа, он может быть преобразован в двоичный файл **access.db** с помощью следующей команды:

 # postmap /etc/postfix/access

Во многих случаях содержимое файлов в каталоге **/etc/postfix** является закомментированной версией соответствующей справочной страницы. Следующие разделы не охватывают файлы **main.cf** и **master.cf**, как это объясняется позже. Они также не охватывают файл **header_checks**, так как это скорее фильтр сообщений.

После внесения каких-либо изменений в файлы конфигурации **Postfix** обычно лучше перезагрузить их в демон с помощью следующей команды:

 # systemctl reload postfix

Файл access Postfix

Файл доступа может быть настроен с ограничениями на **пользователей, хосты и многое другое**. Он включает в себя закомментированную копию связанной справочной страницы, которую также можно вызвать с помощью команды **man 5 access**. Когда в этот файл включены ограничения, они настраиваются в следующей форме:

pattern action

Шаблоны(**pattern**) могут быть настроены несколькими способами. Как подсказывает man 5, вы можете ограничить пользователей такими шаблонами, как

username@example.com

Шаблоны могут быть настроены с **отдельными IP-адресами, сетевыми адресами и доменами**, как в следующих примерах. Обратите внимание на синтаксис, в частности на отсутствие точки в конце 192.168.100 и в начале выражений example.org. Эти выражения включают все системы в сети **192.168.100.0/24 и домен *.example.org.**

192.168.122.50
server1.example.com
192.168.100
example.org

Конечно, такие модели не имеют смысла без действия. Типичные действия включают **REJECT** и **OK**. Следующие примеры строк в файле **/etc/postfix/access** соответствуют формату действия шаблона:

192.168.122.50 OK
server1.example.com OK
192.168.100 REJECT
example.org REJECT

!!!!! EXAM Watch !!!!!!!

Один из способов настройки безопасности *Postfix* на уровне хоста и пользователя - это файл доступа в каталоге */etc/postfix*. Другой способ настройки безопасности на основе хоста - использование богатых правил брандмауэра, как описано в главе 10. Хотя существуют более

сложные методы настройки безопасности на основе пользователей, цели RHCE предполагают, что вы «настраиваете службу для базовой операции».
!!!!!

Postfix файлы canonical и generic

Файлы с именем **canonical** и **generic** в каталоге **/etc/postfix** работают как файлы псевдонимов. Другими словами, когда пользователи перемещаются с места на место или если компания, перемещается из одного домена в другой, канонический файл может облегчить этот переход. Принимая во внимание, что канонический файл применяется к входящей электронной почте из других систем, общий файл применяется к электронной почте, отправляемой в другие системы.

Подобно файлу доступа, параметры в этих файлах следуют шаблону:

pattern result

Простейшей итерацией является следующая, которая пересылает электронную почту, отправленную локальному пользователю, на обычный адрес электронной почты:

michael michael@example.com

Для компаний, которые используют разные домены, следующая строка будет направлять электронную почту, направленную на **michael@example.org**, на **michael@example.com.** Он будет пересылать другие адреса электронной почты **example.org** аналогичным образом.

@example.org @example.com

Не забудьте обработать полученные файлы в базы данных с помощью команд **postmap canonical** и **postmap generic**. Если вы измените перемещенные, транспортные или виртуальные файлы в Каталог **/etc/postfix**, примените команду **postmap** и к этим файлам.

Файл relocated Postfix

Файл **/etc/postfix/relocated** предназначен для хранения информации для пользователей, которые сейчас находятся во внешних сетях, таких как пользователи, покинувшие текущую организацию. Формат аналогичен вышеупомянутым каноническим и общим файлам в одном каталоге. Например, следующая запись может отражать пересылку из локальной корпоративной сети на личный адрес электронной почты:

john.doe@example.com **john.doe@example.net**

Файл Postfix transport

Файл **/etc/postfix/transport** может быть полезен в некоторых ситуациях, когда почта пересылается, например, с промежуточного узла. Например, следующая запись перенаправляет электронную почту, направленную на **домен example.com**, на **SMTP-сервер**, такой как **Postfix**, в системе **server1.example.com**:

example.com smtp:server1.example.com

Файл Postfix virtual

Файл **/etc/postfix/virtual** может пересылать сообщения электронной почты, адресованные обычным способом, например, **elizabeth@example.com**, в учетную запись пользователя в локальной системе. Например, если пользователь **elizabeth** фактически является администратором в системе, следующая запись перенаправляет почту, отправленную на указанный адрес электронной почты, пользователю root-администратора:

elizabeth@example.com root

Файл конфигурации main.cf

Сделайте резервную копию файла конфигурации **main.cf** и откройте его в текстовом редакторе. Есть несколько вещей, которые вы должны настроить в этом файле, чтобы он работал. Когда служба настроена правильно, изменения должны ограничивать доступ к локальной системе и сети. В этом разделе также описывается функция других активных директив в зависимости от версии файла по умолчанию.

Во-первых, очереди **Postfix** включают либо электронную почту, которая еще не отправлена, либо электронную почту, которая была получена. Их можно найти в каталоге **queue_directory**:

queue_directory = /var/spool/postfix

Следующий каталог является стандартным. Он описывает расположение большинства команд **Postfix**.

command_directory = /usr/sbin

Postfix включает в себя значительное количество исполняемых файлов для конфигурации в файле **master.cf**. Директива **daemon_directory** указывает их местоположение:

daemon_directory = /usr/libexec/postfix

Postfix включает записываемые файлы данных в следующем каталоге; обычно он содержит файл **master.lock** с **PID** демона **Postfix**:

data_directory = /var/lib/postfix

Как определено в комментариях к файлу **main.cf**, некоторые файлы и каталоги должны принадлежать пользователю с правами администратора; другие должны принадлежать указанному **mail_owner**. В файле **/etc/groups** вы можете подтвердить выделенную группу с именем **postfix**, а также группу с именем **mail**, которая содержит пользователя **postfix**:

mail_owner = postfix

Хотя **Postfix** работает для локальной системы «**из коробки**», вам нужно сделать больше, чтобы он работал в сети. Для этого вам может потребоваться активировать и изменить следующую директиву **myhostname**, чтобы указать полное доменное имя локальной системы, возвращаемое командой **hostname**. Если это не отличается от имени хоста системы в Интернете, нет необходимости изменять запись.

#myhostname = host.domain.tld

на полное доменное имя, такое как

myhostname = server1.example.com

!!!!! On the job !!!!!
Запись *MX* может быть настроена на официальном *DNS-сервере* для домена, чтобы указать имя хоста *SMTP-сервера*, который принимает электронную почту для этого домена.
!!!!!!

Вам необходимо настроить **SMTP-сервер** для всего доменного имени с помощью директивы **mydomain**. Для этого измените комментарий

#mydomain = domain.tld

для отражения доменного имени локальной сети:

mydomain = example.com

Обычно вы просто раскомментируете следующую директиву **myorigin**, чтобы пометить адреса электронной почты, поступающие с этого сервера **Postfix**, с доменом происхождения. В этом случае исходный домен - **example.com**:

myorigin = $mydomain

По умолчанию следующая активная директива ограничивает область действия службы **Postfix** локальной системой:

#inet_interfaces = all
inet_interfaces = localhost

Для сервера электронной почты, который обрабатывает входящую электронную почту для всего домена, вы обычно меняете активную директиву так, чтобы **Postfix** прослушивал все активные сетевые интерфейсы:

inet_interfaces = all
#inet_interfaces = localhost

Обычно **Postfix** прослушивает в сетях как **IPv4**, так и **IPv6**, основываясь на следующей директиве **inet_protocols**:

inet_protocols = all

Директива **mydestination** указывает системы, обслуживаемые этим сервером **Postfix**. Исходя из предыдущих настроек, следующая директива по умолчанию означает, что принятая почта может быть отправлена на полное доменное имя локальной системы (**server1.example.com**), адрес **localhost** в примере **сеть .com и система localhost**:

mydestination = $myhostname, localhost.$mydomain, localhost

Для сервера **Postfix**, настроенного для локальной сети, вы должны добавить имя локального домена, уже назначенного директиве **mydomain**:

mydestination = $mydomain, $myhostname, localhost.$mydomain, localhost

Задачи **RHCE** требуют настройки нулевого клиента, то есть системы, которая перенаправляет все электронные письма на центральный почтовый сервер. В этом случае вы должны оставить директиву **mydestination** пустой, чтобы указать, что локальная система **Postfix** не является конечным пунктом назначения для каких-либо доменов электронной почты:

mydestination =

Кроме того, вы захотите настроить директиву **mynetworks**, чтобы она указывала на **IP-адрес клиента**, которому доверял этот сервер **Postfix**. Директива с комментариями по умолчанию не указывает на сеть **example.com**, определенную для этой книги:

#mynetworks = 168.100.189.0/28, 127.0.0.0/8

Поэтому для таких систем, как **server1.example.com**, эта директива может быть изменена на

mynetworks = 192.168.122.0/24, 127.0.0.0/8

Если вы настраиваете нулевого клиента, вместо этой директивы должен быть задан **IP-адрес localhost**:

mynetworks = 127.0.0.0/8

Как только изменения, внесенные в файл **main.cf** (и любые другие файлы в каталоге **/etc/postfix**), будут завершены и сохранены, вы можете просмотреть текущие параметры **Postfix**. Для этого выполните следующую команду:

 # postconf

Конечно, большинство этих параметров по умолчанию. Для просмотра параметров определенный файлом **main.cf**, выполните следующую команду:

 # postconf –n

!!!! EXAM Watch !!!!
Вы можете настроить безопасность на основе хоста в Postfix через директива *mynetworks* в Файле */etc/postfix/main.cf*.
!!!!!

РИСУНОК 13-5. Пользовательские настройки *Postfix*, основанные на */etc/posfix/main.cf*

```
[root@Maui postfix]# postconf -n
alias_database = hash:/etc/aliases
alias_maps = hash:/etc/aliases
command_directory = /usr/sbin
config_directory = /etc/postfix
daemon_directory = /usr/libexec/postfix
data_directory = /var/lib/postfix
debug_peer_level = 2
debugger_command = PATH=/bin:/usr/bin:/usr/local/bin:/usr/X11R6/bin ddd $daemon_
directory/$process_name $process_id & sleep 5
html_directory = no
inet_interfaces = localhost
inet_protocols = all
mail_owner = postfix
mailq_path = /usr/bin/mailq.postfix
manpage_directory = /usr/share/man
mydestination = $myhostname, localhost.$mydomain, localhost
mydomain = example.com
myhostname = maui.example.com
mynetworks = 192.168.122.0/24, 127.0.0.0/8
newaliases_path = /usr/bin/newaliases.postfix
queue_directory = /var/spool/postfix
readme_directory = /usr/share/doc/postfix-2.10.1/README_FILES
sample_directory = /usr/share/doc/postfix-2.10.1/samples
sendmail_path = /usr/sbin/sendmail.postfix
setgid_group = postdrop
unknown_local_recipient_reject_code = 550
[root@Maui postfix]#
```

Вывод показан на рисунке 13-5.

Один параметр из вывода **postconf -n важен для аутентификации**. В частности, когда следующая директива добавляется в **файл main.cf**, **Postfix** требует авторизованных имен пользователей и паролей для доступа:

smtpd_sender_restrictions = allow_sasl_authenticated, reject

Кроме того, **Postfix** включает в себя проверку синтаксиса в базовом демоне. Выполните следующую команду, чтобы увидеть, есть ли какие-либо фатальные ошибки в файле **main.cf**:

#postfix check

Файл конфигурации /etc/aliases

Другая директива из файла **/etc/postfix/main.cf** включает хэш базы данных из файла **/etc/aliases**, который обрабатывается в файле **/etc/aliases.db** при перезапуске системы **Postfix**:

alias_maps = hash:/etc/aliases

Файл **/etc/aliases** обычно настраивается для перенаправления электронной почты, отправляемой системным учетным записям, например пользователю **root-администратора**. Как вы можете видеть в конце этого файла, сообщения электронной почты, отправленные пользователю **root**, могут быть перенаправлены на учетную запись обычного пользователя:

 # root marc

Хотя в этом файле имеется ряд дополнительных директив, они выходят за рамки базовой конфигурации, связанной с целями **RHCE**. После внесения изменений вы можете и должны обработать этот файл в соответствующей базе данных с помощью команды **newaliases**.

Проверьте текущую конфигурацию Postfix

Как отмечалось в предыдущих главах, команда **telnet** - отличный способ проверить текущее состояние службы в локальной системе. Исходя из конфигурации **Postfix** по умолчанию, активная версия этой службы должна прослушивать **порт 25**. В этом случае команда **telnet localhost 25** должна возвращать сообщения, подобные следующим:

Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 server1.example.com ESMTP Postfix

Если в локальной системе включена сеть IPv6, **адрес обратной связи (loopback address) IPv4 (127.0.0.1)** будет заменен обычным адресом обратной **связи IPv6 (:: 1).** Команда **quit** может быть использована для выхода из этого соединения. Но пока не уходите. Введите команду **EHLO localhost** и нажмите ввод; **EHLO** - это расширенная команда **HELO**, которая возвращает основные параметры **SMTP-сервера**.

EHLO localhost
250-maui.example.com
250-PIPELINING
250-SIZE 10240000
250-VRFY
250-ETRN
250-ENHANCEDSTATUSCODES
250-8BITMIME
250 DSN

Для наших целей самая важная информация - это то, чего не хватает. На этом сервере аутентификация не требуется. Когда аутентификация правильно настроена в Postfix, вы также увидите следующую строку в выводе:

250-AUTH GSSAPI

Настроить аутентификацию Postfix

Когда аутентификация настроена в **Postfix**, могут применяться ограничения пользователя. Однако, поскольку в стандартном файле конфигурации **main.cf** нет подсказок, вам придется обратиться к документации **Postfix** для получения подсказок. Как предлагается в Главе 3, большинство пакетов содержат некоторый уровень документации в каталоге **/usr/share/doc**. К счастью, документация **Postfix** в этом каталоге довольно обширна. В RHEL 7 вы сможете найти эту документацию в подкаталоге **postfix-2.10.1/.**

Директивы, которые нужно добавить в файл **main.cf** для настройки аутентификации, показаны в файле **README-Postfix-SASL-RedHat.txt** в этом каталоге. Отрывок ключа показан на **рисунке 13-6**.

Для первого перечисленного шага достаточно скопировать четыре директивы, указанные в конецt файла **main.cf**. Первый включает **аутентификацию SASL** для соединений **Postfix**:

РИСУНОК 13-6 Указания по настройке аутентификации Postfix

```
Quick Start to Authenticate with SASL and PAM:
----------------------------------------------

If you don't need the details and are an experienced system
administrator you can just do this, otherwise read on.

1) Edit /etc/postfix/main.cf and set this:

smtpd_sasl_auth_enable = yes
smtpd_sasl_security_options = noanonymous
broken_sasl_auth_clients = yes

smtpd_recipient_restrictions =
  permit_sasl_authenticated,
  permit_mynetworks,
  reject_unauth_destination

2) Turn on saslauthd:

   /sbin/chkconfig --level 345 saslauthd on
   /sbin/service saslauthd start

3) Edit /etc/sysconfig/saslauthd and set this:

   MECH=pam

4) Restart Postfix:

   /sbin/service postfix restart
```

smtpd_sasl_auth_enable = yes

Следующая директива отключает анонимную аутентификацию:

smtpd_sasl_security_options = noanonymous

Следующая директива разрешает аутентификацию от нестандартных и устаревших клиентов, таких как **Microsoft Outlook Express:**

broken_sasl_auth_clients = yes

Следующая строка разрешает аутентифицированным пользователям, предоставляет доступ из сетей, настроенных с помощью директивы **mynetworks**, и отклоняет места назначения, отличные от сервера **Postfix**:

**smtpd_recipient_restrictions = permit_sasl_authenticated,
permit_mynetworks, reject_unauth_destination**

Настройте Postfix как SMTP-сервер для домена

Директивы, необходимые для настройки **Postfix** для приема входящей электронной почты из других систем, ранее были показаны в описании файла **main.cf**. Однако это обсуждение было более полным описанием этого файла. В этом разделе кратко изложены минимальные требования для настройки **Postfix** для приема входящих сообщений электронной почты от других систем. Учитывая сервер **Postfix**, настроенный в системе **maui.example.com**, в сети **192.168.122.0/24** вы внесете изменения, показанные в **таблице 13-5**, в файл **main.cf** в каталоге **/etc/postfix**.

Каждый из этих параметров заменяет либо комментарий, либо активную директиву по умолчанию файл **/etc/postfix/main.cf**. Например, вы должны хотя бы закомментировать следующую директиву:

#inet_interfaces = localhost

ТАБЛИЦА 13-5 Конфигурация Postfix в качестве SMTP-сервера для example.com

Postfix Parameter	Описание
myhostname = maui.example.com	Определяет имя хоста системы
mydomain = example.com	Устанавливает имя локального домена
myorigin = $mydomain	Указывает домен, с которого будут отображаться локальные сообщения электронной почты
mydestination = $myhostname, localhost.$mydomain, localhost, $mydomain	Перечисляет домен, для которого эта машина является местом назначения
inet_interfaces = all	Сообщает **Postfix** прослушивать все интерфейсы
mynetworks = 192.168.122.0/24, 127.0.0.0/8	Перечисляет диапазон IP-адресов доверенных SMTP-клиентов.

Настройте Postfix как нулевой клиент

В этом разделе рассматриваются минимальные требования для настройки **Postfix**, для целей RHCE, «пересылать всю электронную почту на центральный почтовый сервер». Интеллектуальный хост обеспечивает это Функциональность и работает как обычный **SMTP-сервер**, за исключением пересылки всей электронной почты через **второй SMTP-сервер**. Местоположение промежуточного узла может быть указано с помощью директивы **relayhost**. Например, если удаленный промежуточный узел - **outsider1.example.org**, вы добавили бы следующую директиву в файл **/etc/postfix/main.cf**:

relayhost = outsider1.example.org

Конфигурация нулевого клиента еще более ограниченная, чем интеллектуальный хост. Как и в случае конфигурации промежуточного узла, все электронные письма пересылаются на центральный почтовый сервер. Кроме того, сообщения электронной почты не принимаются для локальной доставки. Соответствующие настройки конфигурации показаны в **таблице 13-6**.

ТАБЛИЦА 13-6 Конфигурация Postfix в качестве нулевого клиента

Параметры Postfix	Описание
myhostname = server1.example.com	Определяет имя хоста системы
mydomain = example.com	Устанавливает имя локального интернет-домена
myorigin = server1.example.com	Сообщает Postfix, что сообщения электронной почты должны отображаться для отправки с домена server1.example.com
mydestination =	Настраивает систему как нулевого клиента (другими словами, как машину, которая не является пунктом назначения для какого-либо домена)
local_transport = error: local mail delivery is disabled	Отключает доставку электронной почты в локальную систему
inet_interfaces = localhost	Направляет **Postfix** слушать только интерфейс **localhost**
relayhost = maui.example.com	Пересылает все электронные письма на хост **maui.example.com**
mynetworks = 127.0.0.0/8	Перечисляет диапазон IP-адресов доверенных SMTP-клиентов.

ЦЕЛЬ СЕРТИФИКАЦИИ 13.05

Цели и инициаторы iSCSI

Соответствующая цель **RHCE** этого раздела - «**настроить систему как taget iSCSI или инициатор, который постоянно монтирует taget iSCSI»**. Инициатор **iSCSI** является клиентом. Цель **iSCSI** - это общее хранилище на сервере, которое обменивается данными с клиентом через **TCP-порт 3260.**

Протокол iSCSI инкапсулирует и **доставляет команды SCSI по IP-сети**. После настройки сервера и клиента у вас будет **доступ к LUN хранилищу** к цели **iSCSI**; этот **LUN** будет выглядеть как еще один жесткий диск **SCSI** на клиенте.

Установите программную цель iSCSI

Сегодня многие массивы хранения поддерживают протокол **iSCSI**. Однако для проведения экзамена **RHCE** вам необходимо узнать, как настроить **сервер Linux в качестве цели iSCSI** (то есть **сервера хранения iSCSI**). Конечно, задержка и время отклика, вероятно, будут медленнее, чем в массиве хранения **iSCSI** корпоративного класса, но это зависит от многих факторов, включая тип дисков и пропускную способность сети.

!!!! On the job !!!!!
В производственном развертывании iSCSI вы можете рассмотреть возможность включения «больших кадров» на всех целевых объектах, инициаторах и коммутаторах Ethernet в структуре iSCSI. Jumbo-кадры - это Ethernet-кадры с большим размером MTU (обычно 9000 байт), и поэтому они обычно обеспечивают лучшую пропускную способность, чем стандартный 1500-байтовый MTU. Чтобы включить Jumbo-кадры на сетевой карте в RHEL 7, добавьте директиву MTU = 9000 в соответствующий файл конфигурации ifcfg- * в каталоге /etc/sysconfig/network-scripts.
!!!!!

Один из способов установки цели **iSCSI** - с помощью пакета **targetcli**. Установите его как показано далее:

 # yum install targetcli

В пакет входит команда **targetcli**, которая запускает удобную для пользователя оболочку конфигурации, которая проведет вас через все шаги по развертыванию **цели iSCSI**. После запуска оболочки **targetcli** введите команду **ls**. Вы увидите результат, показанный на **рисунке 13-7**.

Из оболочки **targetcli** вы можете переходить к различным разделам конфигурации с помощью команды **cd**, как в файловой системе. Команда **ls** отображает содержимое текущего раздела, в то время как справка предоставляет полезный экран контекстной справки. Как и в обычной оболочке, вы можете использовать завершение табуляции для заполнения частично введенных команд или аргументов.

РИСУНОК 13-7 Административная оболочка targetcli

```
[root@server1 ~]# targetcli
targetcli shell version 2.1.fb34
Copyright 2011-2013 by Datera, Inc and others.
For help on commands, type 'help'.

/> ls
o- / ......................................................................... [...]
  o- backstores .............................................................. [...]
  | o- block .................................................. [Storage Objects: 0]
  | o- fileio ................................................. [Storage Objects: 0]
  | o- pscsi .................................................. [Storage Objects: 0]
  | o- ramdisk ................................................ [Storage Objects: 0]
  o- iscsi ............................................................ [Targets: 0]
  o- loopback ......................................................... [Targets: 0]
/>
```

Настройте Backstore(резервное хранилеще)

Как показано на **Рисунке 13-7**, первый шаг состоит в настройке резервного хранилища, то есть устройства резервного хранилища, которое впоследствии будет экспортировано клиентам **iSCSI**. Если вы настроили виртуальные машины, как предложено **в главе 1**, у вас должно быть достаточно свободного места на локальном диске, чтобы создать новый логический том, который будет выделен для области хранилища **iSCSI**.

Например, войдите на сервер **server1.example.com** и создайте новый логический том размером 1ГБ в группе томов **rhel_server1** по умолчанию, которая была создана во время установки операционной системы (замените соответственно, если имя вашей группы томов отличается):

```
# lvcreate -L 1G -n backstore rhel_server1
Logical volume "backstore" created
```

Для устройства резервного хранилища вы можете использовать любое блочное устройство, такое как логический том, раздел диска или даже целый диск. Но это еще не все. Как показано на рисунке 13-7, **targetcli** поддерживает не только блочные устройства в качестве резервного хранилища, но также **image** файлы (**fileio**), физические диски **SCSI** в сквозном (**pass-through**) режиме (**pscsi**) и временные файловые системы в памяти (**ramdisk**). Для целей этого раздела мы будем использовать блочное устройство.

Когда у вас есть готовое к настройке блочное устройство, вернитесь в оболочку **targetcli** и создайте объект блочного хранилища:

```
/> cd backstores/block
/backstores/block> create disk1 /dev/rhel_server1/backstore
Created block storage object device1 using /dev/rhel_server1/backstore
```

Эта команда создания указывает **targetcli** использовать том **/dev/rhel_server1/backstore** в качестве объекта хранилища блоков с именем **disk1**.

Настройте квалифицированное имя iSCSI

Из оболочки **targetcli** перейдите к пути **/iscsi**:

```
/backstores/block> cd /iscsi
/iscsi>
```

Введите команду **help**, чтобы отобразить список доступных параметров. Следующий шаг состоит в создании квалифицированного имени **iSCSI Qualified Name (IQN)**. Это уникальная строка, которая идентифицирует инициатор или цель **iSCSI**, например:

iqn.2015-01.com.example: сервер1-disk1

IQN должен соответствовать определенному формату. Он должен начинаться со строки «**iqn**.», За которой следует год и месяц (**ГГГГ-ММ**), в которых организация зарегистрировала свой публичное доменное имя.

Далее идет домен организации в обратном порядке, за которым следует необязательная строка, разделенная столбцами.

Вернемся к оболочке **targetcli**, давайте создадим **IQN**:

```
/iscsi> create iqn.2015-01.com.example:server1-disk1
Created target iqn.2015-01.com.example:server1-disk1.
Created TPG 1.
/iscsi>
```

Как указано в выходных данных предыдущей команды, **targetcli** создал **IQN** для цели и нового объекта **TPG 1. TPG** - это целевая группа портала. Его цель - связать вместе несколько частей конфигурации, как вы увидите в следующем разделе.

Настройте целевую группу портала (Target Portal Group)

Если вы выполнили шаги настройки, показанные до сих пор, введите команду **ls** из оболочки **targetcli**. Вы увидите результат, показанный на **рисунке 13-8**.

Как видно на рисунке, оболочка **targetcli** содержит новые пункты меню под строками **IQN** и **TPG**. Следуя порядку, показанному на **рисунке 13-8**, мы выполним следующие шаги:

1. Мы настроим **список контроля доступа (ACL)**, чтобы разрешить доступ к цели только от определенного клиента.
2. Мы создадим номер логического устройства **(logical unit number) (LUN)** для текущего устройства хранилища.
3. Мы определим портал, то **есть IP-адрес** и, необязательно, пользовательский порт, который целевой объект **iSCSI** будет прослушивать для соединений.

Чтобы настроить **ACL** и ограничить доступ к **IQN** определенного инициатора **iSCSI**, введите следующие команды:

/iscsi> cd iqn.2015-01.com.example:server1-disk1/tpg1/acls
/iscsi/iqn.20...sk1/tpg1/acls> create iqn.2015-01.com.example:tester1
Created Node ACL for iqn.2015-01.com.example:tester1
/iscsi/iqn.20...sk1/tpg1/acls>

РИСУНОК 13-8 Настройка TPG из оболочки targetcli

```
/iscsi> ls
o- iscsi ..................................................... [Targets: 1]
  o- iqn.2015-01.com.example:server1-disk1 ..................... [TPGs: 1]
    o- tpg1 ..................................... [no-gen-acls, no-auth]
      o- acls ..................................................... [ACLs: 0]
      o- luns ..................................................... [LUNs: 0]
      o- portals ................................................. [Portals: 0]
/iscsi>
```

Затем перейдите к разделу **LUNs** и свяжите номер **LUN** с ранее созданным устройством **backstore**:

/iscsi/iqn.20...sk1/tpg1/acls> cd ../luns
/iscsi/iqn.20...sk1/tpg1/luns> create /backstores/block/disk1 0
Created LUN 0.
Created LUN 0->0 mapping in node ACL iqn.2015-01.com.example:tester1
/iscsi/iqn.20...sk1/tpg1/luns>

Наконец, перейдите к разделу портала и создайте новый портал **iSCSI** для прослушивания локального **IP-адреса (192.168.122.50 в этом примере)**. Если вы не укажете **порт TCP**, по умолчанию **targetcli** будет использовать порт **TCP 3260**:

/iscsi/iqn.20...sk1/tpg1/luns> cd ../portals
/iscsi/iqn.20...tpg1/portals> create 192.168.122.50
Using default IP port 3260
Created network portal 192.168.122.50:3260
/iscsi/iqn.20...tpg1/portals>

Это завершает настройку **TPG**. Введите **ls /**, чтобы показать полную конфигурацию **цели iSCSI**, как показано **на рисунке 13-9**. Затем введите команду **exit**, чтобы закрыть оболочку **targetcli**. Конфигурация сохраняется автоматически.

Все системные службы должны быть настроены для запуска при загрузке. Для целевой службы **iSCSI** нет исключений, поэтому вы должны убедиться, что она настроена на запуск при следующем включении машины. Это можно сделать, выполнив следующие команды:

 # systemctl enable target

 # systemctl start target

РИСУНОК 13-9 Конфигурация цели iSCSI

```
/iscsi/iqn.20.../tpg1/portals> ls /
o- / ..................................................................... [...]
  o- backstores ........................................................... [...]
  | o- block ..................................... [Storage Objects: 1]
  | | o- disk1 ..... [/dev/rhel_server1/backstore (1.0GiB) write-thru activated]
  | o- fileio .................................... [Storage Objects: 0]
  | o- pscsi ..................................... [Storage Objects: 0]
  | o- ramdisk ................................... [Storage Objects: 0]
  o- iscsi ..................................................... [Targets: 1]
  | o- iqn.2015-01.com.example:server1-disk1 ..................... [TPGs: 1]
  |   o- tpg1 ..................................... [no-gen-acls, no-auth]
  |     o- acls ................................................... [ACLs: 1]
  |     | o- iqn.2015-01.com.example:tester1 ................. [Mapped LUNs: 1]
  |     |   o- mapped_lun0 ......................... [lun0 block/disk1 (rw)]
  |     o- luns ................................................... [LUNs: 1]
  |     | o- lun0 ................. [block/disk1 (/dev/rhel_server1/backstore)]
  |     o- portals ................................................ [Portals: 1]
  |       o- 192.168.122.50:3260 ..................................... [OK]
  o- loopback .................................................. [Targets: 0]
/iscsi/iqn.20.../tpg1/portals>
```

Не забудьте разрешить соединения через локальный брандмауэр. По умолчанию целевая служба **iSCSI использует TCP-порт 3260:**

 # firewall-cmd --permanent --add-port=3260/tcp
 # firewall-cmd --reload

Подключение к удаленному хранилищу iSCSI

В этом разделе мы настроим виртуальную машину **tester1.example.com** для монтирования **LUN**, экспортируемого целевым объектом **iSCSI**, определенным в предыдущем разделе. Для настройки **клиента iSCSI** вам потребуются пакеты **iscsi-initiator-utils**, а также любые зависимости:

 # yum install iscsi-initiator-utils

!!!!! On the job !!!!!!
Пакет iscsi-initiator-utils реализует программный инициатор iSCSI. Однако сегодня большинство сетевых адаптеров предоставляют аппаратные функции инициатора iSCSI. Конфигурация аппаратного инициатора iSCSI зависит от производителя карты и, как таковая, зависит от разных производителей и моделей карт.
!!!!!!!

Затем настройте **IQN** для инициатора. Это определено в файле **/etc/iscsi/initiatorname.iscsi.** Отредактируйте содержимое и введите **определённый IQN**:

InitiatorName=iqn.2015-01.com.example:tester1

Если вы настроили **ACL** на **цели iSCSI, IQN** клиента должен соответствовать **IQN**, определенному в **ACL**; в противном случае клиенту не будет предоставлен доступ к цели.

Затем включите службу **iscsi** на клиенте и убедитесь, что она запускается при следующей загрузке:

 # systemctl start iscsi
 # systemctl enable iscsi

Вы использовали утилиту **iscsiadm** для обнаружения доступных целей **iSCSI**. Один метод с помощью следующей команды:

 # iscsiadm -m discoverydb -t st -p 192.168.122.50 -D

Для интерпретации эта команда **iscsiadm** запрашивает цели **iSCSI**. Он работает в режиме базы данных обнаружения (**discoverydb**) **mode** (**-m**), где тип обнаружения **type (-t)** запрашивает в команде **sendtargets** (или **st**) была отправлена цели **iSCSI**, определенной на портале **portal (-p),** прослушивающем отмеченный **IP-адрес**, чтобы обнаружить **discover (-D) LUN** общего хранилища.

В случае успеха вы увидите вывод, подобный следующему:

192.168.122.50:3260,1 iqn.2015-01.com.example:server1-disk1

Чтобы использовать только что обнаруженную цель, вам нужно выполнить следующую команду:

 # iscsiadm -m node -T iqn.2015-01.com.example:server1-disk1 –l

Эта команда работает в режиме узла **(node mode) (-m)** для входа в систему **(log in) (-l)** целевого **IQN (-T) iqn.2015-01.com.example: сервер1-disk1**. В случае успеха вы сможете увидеть дополнительное дисковое устройство хранения запустив команду **fdisk –l.**

После этого вы сможете управлять общим хранилищем, как если бы это был новый жесткий диск в локальной системе. Файл устройства жесткого диска будет отображаться в файле **/var/log/messages** с такой информацией, как следующая, которая указывает на файл устройства **/dev/sdc**:

Sep 25 20:22:15 tester1 kernel sd 6:0:0:0: [sdc] Attached SCSI disk

Затем вы можете создавать разделы и многое другое на новом диске **/dev/sdc**, как если бы это был локальный диск, основываясь на методах, описанных в **главе 6**. Конечно, «постоянное подключение», как описано в соответствующей задаче **RHCE** требует, чтобы вы убедились, что **служба iSCSI** запускается при следующей перезагрузке системы.

Чтобы убедиться в наличии фактического монтирования, вам также может понадобиться настроить раздел, который фактически монтируется в файле **/etc/fstab**. На практике фактический файл устройства для диска **iSCSI** может меняться при каждой перезагрузке. Поэтому такие монтирования должны быть настроены с номерами универсального уникального идентификатора (**UUID**), описанными в **главе 6**.

ЦЕЛЬ СЕРТИФИКАЦИИ 13.06

Служба сетевого времени

Настройка **NTP** в качестве клиента и сервера по умолчанию рассматривается в **главе 5**. В отличие от этого, здесь описывается, который синхронизирует время с использованием пиров NTP. Тем не менее, вам нужно знать, как **настройка NTP в качестве сервера** защитить NTP так же, как вы защищаете другие сетевые сервисы, такие как Samba и NFS.

Чтобы **разрешить NTP работать** в качестве сервера, необходимо **разрешить доступ через UDP-порт 123**. Это может быть достигнуто путем добавления **службы ntp в соответствующую зону firewalld.**

Файл конфигурации NTP-сервера

Как обсуждалось **в главе 5**, конфигурация времени зависит от часового пояса, в котором Ссылка на символическую ссылку **/etc/localtime**, а также **серверы NTP**, настроенные в файле **/etc/ntp.conf (или в файле /etc/chrony.conf, если используется chronyd).** Теперь пришло время настроить один из этих серверов **NTP**. Мы сосредоточимся на ntpd, так как это стандартный демон **NTP** для систем, которые всегда подключены к сети.

Файл конфигурации **/etc/ntp.conf** по умолчанию начинается с директивы **driftfile**, которая отслеживает отклонение и ошибки в системных часах локальной системы:

driftfile /var/lib/ntp/drift

Существуют также ограничительные директивы, которые могут помочь защитить NTP-сервер. Эта директива работает с IPv4 и с сетями IPv6, как показано здесь:

restrict default nomodify notrap nopeer noquery
restrict 127.0.0.1
restrict ::1

Параметры директивы **restrict** могут быть описаны следующим образом:

default Относится к соединениям по умолчанию из других систем; может быть дополнительно ограничено другими ограничительными директивами.
nomodify Запрещает запросы, которые пытаются изменить конфигурацию локального NTP-сервера.
notrap Запрещает службу прерывания управляющего сообщения; Вы можете удалить эту опцию, чтобы включить удаленное ведение журнала.
nopeer Останавливает доступ с потенциальных одноранговых NTP-серверов.
noquery Игнорирует запросы.

Тем не менее, эти ограничения, в сочетании, хороши только для **клиента NTP**. Чтобы настроить **NTP-сервер**, в частности тот, который «синхронизирует время с использованием других пиров NTP», вы должны удалить хотя бы директиву **nopeer** из списка ограничений. Некоторые **NTP**-серверы могут нуждаться в синхронизации с вашими, что возможно, если вы удалите запрос и из списка.

Следующие две директивы ограничения ограничивают доступ к локальному NTP-серверу в локальной системе. Вы должны распознать адреса обратной связи IPv4 и IPv6 по умолчанию здесь:

restrict 127.0.0.1
restrict ::1

Конечно, при настройке **NTP-сервера** для других клиентов вы захотите ослабить это ограничение. Следующий комментарий включает сетевой адрес в требуемом формате. Поэтому, чтобы настроить **NTP-сервер** для **сети 192.168.122.0/24**, вы должны изменить директиву **restrict** на

restrict 192.168.122.0 mask 255.255.255.0 notrap nomodify

Для базовой конфигурации «NTP-сервер по умолчанию» никаких дополнительных изменений не требуется. Конечно, локальный NTP-сервер также должен быть настроен как клиент для управления **NTP-серверами**. Замените имя хоста для фактических серверов **NTP** в вашей сети на следующее:

server 0.rhel.pool.ntp.org iburst
server 1.rhel.pool.ntp.org iburst
server 2.rhel.pool.ntp.org iburst
server 3.rhel.pool.ntp.org iburst

И чтобы повторить ссылку из целей **RHCE**, другая ссылка на **peers**. Соответствующая директива является пэром.

Чтобы протестировать директиву на одном **NTP-сервере**, вы можете настроить этот компьютер на соединение с другим хостом, как показано здесь:

peer server1.example.com

В качестве альтернативы, вы могли бы дать имя хоста одноранговому серверу NTP, возможно, в корпоративной сети, возможно, в сети, которая была настроена во время экзамена.

Пределы безопасности на NTP

Как только что описано, директива **restrict** из файла конфигурации **/etc/ntp.conf** может использоваться для ограничения доступа к локальному **NTP-серверу**, но это предполагает открытый **порт 123**. Пределы безопасности также могут относиться к настроенным брандмауэрам. Просто имейте в виду, что соответствующее правило брандмауэра для NTP открывает **порт 123. UDP (не TCP).** Это можно настроить, добавив службу **ntp** в соответствующую зону брандмауэра, например:

 # firewall-cmd --permanent --add-service = ntp
 # firewall-cmd --reload

Чтобы проверить соединение с **NTP-сервером**, выполните команду **ntpq -p hostname**. Эта команда ищет пиры, перечисленных в файле **/etc/ntp.conf**. Если сервер работает, вы увидите нечто похожее на следующий вывод команды **ntpq -p localhost**, показанной на **рисунке 13-10**.

Знак * перед именем хоста или IP-адресом указывает, что текущий одноранговый узел или сервер **NTP** используется в качестве первичной ссылки, тогда как знак + указывает дополнительные одноранговые узлы, обозначенные как приемлемые для синхронизации. Конечно, если команда **ntpq** работает из удаленной системы, используя локальное имя хоста или IP-адрес, вы убедились, что удаленный NTP-сервер работает.

РИСУНОК 13-10 Состояние NTP-сервера, проверяется командой ntpq –p

```
[root@server1 ~]# ntpq -p
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
+time2.mediainve 131.188.3.220    2 u   21   64    3   26.719    1.814  12.969
+stz-bg.com      192.53.103.104   2 u   19   64    3   54.418    6.646  28.389
*ntp2.litnet.lt  .GPS.            1 u   18   64    3   48.583    2.647  22.977
+betelgeuse.retr 193.62.22.90     2 u   17   64    3    2.299    0.548  25.659
[root@server1 ~]#
```

!!!!! EXAM watch !!!!!!!!
Поскольку NTP является службой на основе UDP, команда telnet не будет проверять работу этой службы. Вы можете проверить локальное состояние службы NTP с помощью команд ntpq -p и nmap -sU -p123.
!!!!!!!

СЦЕНАРИЙ И РЕШЕНИЕ.	
Вам необходимо **настроить DNS-сервер** только для кэширования для локальной сети.	Используйте файл **named.conf** по умолчанию; измените директивы **listen-on и allow-query**.
Вам нужно настроить **DNS-сервер** только для кэширования для пересылки запросов в другие места.	Используйте файл **named.conf,** настроенный как сервер имен только для кэширования; добавьте директиву **forwarders**, указывающую на нужный **DNS-сервер**.
Вам предлагается **настроить SMTP-сервер для сети 192.168.0.0/24**.	Используйте сервер **Postfix** по умолчанию; измените директивы **myhostname**, **mydomain**, **myorigin**, **mydestination**, **inet_interfaces** и **mynetworks** в **/etc/postfix/main.cf**.
Вас попросят настроить **Postfix** как нулевого клиента.	Установите **relayhost** на удаленный сервер для пересылки электронной почты; изменить **mydestination** и **local_transport**; и ограничить доступ к серверу с помощью директив **inet_interfaces** и **mynetworks**.
Вам предлагается разрешить доступ только к **SMTP-серверу для user1, user2 и user3.**	Создать отмеченных пользователей с помощью **shell /sbin/nologin** по умолчанию.
Вам нужно установить **NTP-сервер** как одноранговый.	Модифицируйте файл **ntp.conf**, чтобы указать хост или IP-адрес одноранговой NTP-сервера с помощью директивы **peer**. Добавьте директиву **restrict** без опции **nopeer**, чтобы разрешить доступ к нужному хосту.

РЕЗЮМЕ СЕРТИФИКАЦИИ

DNS предоставляет базу данных доменных имен и IP-адресов, которые помогают хостам преобразовывать имена хостов в IP-адреса в различных сетях, включая Интернет. Это распределенная база данных, в которой каждый администратор отвечает за свою зону полномочий, например **mheducation.com. DNS-сервер** по умолчанию использует именованный демон, основанный на домене

имен в Интернете Беркли **(Berkeley Internet Name Domain)** (**BIND**). Другие альтернативы, такие как **Unbound resolver**, также доступны.

Существует четыре основных **типа DNS-серверов: главный, подчиненный (или вторичный), только для кэширования и только для пересылки**. Цели RHCE специально исключают главные и подчиненные серверы имен. Файл **/etc/named.conf** по умолчанию создается для конфигурации **DNS-сервера только для кэширования**. Сервер имен только для пересылки использует директивы только перенаправления и пересылки в файле **named.conf**. В любом случае вы должны сконфигурировать директивы **listen-on и allow-query** для поддержки доступа из локальной системы и желаемых сетей. Чтобы **протестировать DNS-сервер**, используйте такие команды, как **rndc status**, **dig** и **host**.

Red Hat включает в себя два сервера, связанных с протоколом **SMTP: Postfix и sendmail**. **Postfix** является сервером **SMTP** умолчанию и его несколько проще настроить, чем **sendmail**. Различные файлы конфигурации **Postfix** можно найти в каталоге **/etc/postfix**. Ограничения пользователя и хоста могут быть настроен в файле доступа. Несколько других файлов относятся к перенаправленной или переименованной электронной почте учетные записи или домены. Вам необходимо изменить директивы конфигурации **Postfix** в **/etc/postfix/** в файле **main.cf**, включая **myhostname, mydomain, myorigin, mydestination, inet_interfaces**, и **mynetworks**. Директива **relayhost** может помочь настроить переадресацию на промежуточный узел. Если вам нужно настроить пустой (нулевой) клиент, вам также нужно установить директиву **local_transport**, чтобы избежать доставки электронных писем в локальную систему.

Протокол **iSCSI** эмулирует шину **SCSI** по **IP-сети**. С помощью команды **shell targetcli**, вы можете интерактивно настроить хост **Linux,** как программную цель **iSCSI** для экспорта локального хранилища в удаленные инициаторы **iSCSI**. Инициаторы **iSCSI** могут обнаруживать удаленные цели и входить в них, используя команда **iscsiadm**.

Наконец, чтобы настроить **NTP-сервер** для сети, вам нужно изменить **/etc/ntp.conf** файл. Директива **restrict** должна быть изменена, чтобы указать сетевой адрес. Поддерживать пирсы, предложенные в целях **RHCE**, вам также нужна директива **restrict** без **noquery** и (наиболее важные) варианты **nopeer**. Затем, чтобы настроить другие системы в качестве пиров, вы используете формат **peer hostname**.

Пару минут проверки

Вот некоторые из ключевых моментов целей сертификации в главе 13.

Введение в службы доменных имен

- DNS основан на домене имен в Интернете Berkeley (**BIND**) с использованием демона **named**.
- Пакеты ключей включают в себя **bind-chroot**, который добавляет безопасность, поддерживая **DNS** в **chroot jail** и **bind-utils**, которые включают утилиты команд, такие как **dig** и **host**.
- Четыре основных типа **DNS-серверов**: главный, подчиненный (вторичный), только для кэширования и только пересылка (**master, slave (secondary), caching-only,** и **forwarding-only**). Цели RHCE требуют охвата только **сервисы DNS кэширования**.

Минимальные конфигурации DNS-сервера

- Критичные файлы конфигурации **BIND** включают **/etc/named.conf** и файлы в каталоге **/var/named**.
- Стандартный файл **/etc/named.conf** настроен для сервера имен только для кэширования, ограниченный локальной системой. Изменения в директивах **listen-on** и **allow-query** могут включить доступ с **DNS-клиентов** по сети.
- Сервер имен переадресации требует директивы только перенаправления и пересылки, которая указывает **IP-адреса** удаленных **DNS-серверов**.
- Программа **Unbound** предоставляет альтернативу **BIND** для настройки только безопасного кэширования и переадресация сервера имен.

Разнообразие агентов электронной почты

- **Postfix** является агентом пересылки почты по умолчанию в **RHEL 7**.
- Информация о почтовом сервере записывается в файл **/var/log/maillog**.

Конфигурация Postfix

- Сервер **Postfix** можно настроить с помощью файлов конфигурации в каталоге **/etc/postfix**. Основной файл конфигурации - это файл **main.cf**.
- Вы можете настроить псевдонимы электронной почты в **/etc/aliases**.
- Вы можете настроить различные виды пересылки электронной почты в таких файлах, как канонические, общие, и переместились, все в каталоге **/etc/postfix**.
- Директива **relayhost** может использоваться для установки соединения с промежуточным узлом.
- Чтобы настроить **Postfix** в качестве нулевого клиента, необходимо запретить доставку электронной почты в локальную систему в директиве **local_transport**.
- Вы можете протестировать стандартную конфигурацию **Postfix** из локальной системы с помощью команды **telnet localhost 25**.

Цели(targets) и инициаторы (Initiators) iSCSI

- Вы можете настроить **цель iSCSI**, активировав целевую службу и запустив управляющею оболочку **targetscli**.
- Чтобы настроить цель **iSCSI**, вам нужно определить устройство хранения, установить **IQN**, определить **LUN**, создайте портал и (необязательно) определите **ACL**.
- Для настройки клиента **iSCSI** необходим пакет **iscsi-initiator-utils**, который может быть использована для обнаружения и входа в систему для **целей iSCSI** с помощью команды **iscsiadm**.
- Чтобы убедиться, что соединение **iSCSI** выдерживает перезагрузку, вам нужно активировать сервис **iscsi**.

Служба сетевого времени

- Стандартный файл **конфигурации NTP, /etc/ntp.conf**, настраивает клиента с ограниченным доступом в локальной системе.
- Стандартная директива **restrict** в файле **ntp.conf** по умолчанию доступна для открытого доступа к системам в указанной сети. Вам также необходимо разрешить трафик **NTP** через локальный брандмауэр.
- Цели **RHCE** предполагают связь с peep; такие соединения могут быть настроен с помощью одноранговой директивы.

САМОПРОВЕРКА

Следующие вопросы помогут оценить ваше понимание материала, представленного в этом глава. Поскольку на экзаменах Red Hat нет вопросов с несколькими вариантами ответов, нет вопросов с несколькими вариантами ответов появляются в этой книге. Эти вопросы исключительно проверяют ваше понимание главы. Это нормально, если у вас есть другой способ выполнения задачи. Получение результатов, а не запоминание пустяков, это то, что рассчитывают на Red Hat экзаменах. На многие вопросы может быть более одного ответа.

Введение в службы доменных имен

1. Назовите два пакета, которые предоставляют услуги **DNS на RHEL 7.**

_____

Минимальные конфигурации DNS-сервера

2. Чтобы настроить связь **DNS** через **порт 53**, какие изменения вы бы внесли в брандмауэр для поддержки доступа других клиентов к локальному **DNS-серверу**?

_____

3. Какой файл содержит базовый шаблон для **DNS-сервера BIND** только для кэширования имен?

_____

4. Какая команда гарантирует, что служба **BIND DNS** запускается при следующей загрузке **Linux** в задачах по умолчанию?

_____

Разнообразие агентов электронной почты

5. Перечислите два примера MTA, доступных на RHEL 7.

_____

6. Какую команду можно использовать для переключения между установленными службами **Postfix** и **sendmail**?

_____

Конфигурация Postfix

7. Как бы вы изменили следующую директиву в **/etc/postfix/main.cf**, чтобы открыть **Postfix** для всей системы?

inet_interfaces = localhost

_____

8. Если вы используете **/etc/aliases** для пересылки электронной почты, какая команда обрабатывает эти файлы в подходящий файл базы данных для **Postfix**?

_____

9. Какая директива в файле **main.cf** используется для указания домена, обслуживаемого сервером **Postfix**?

_____

Цели и инициаторы iSCSI

10. Какая служба должна быть запущена при перезагрузке на правильно настроенной цели iSCSI?

_____

11. Какую командную утилиту можно использовать для настройки цели **iSCSI**?

_____

Служба сетевого времени

12. Введите директиву, подходящую для **/etc/ntp.conf**, которая ограничивает доступ к сети **192.168.0.0/24**.

_____

ВОПРОСЫ ЛАБОРОТОРНОЙ РАБОТЫ

Некоторые из этих лабораторий включают упражнения по настройке. Вы должны делать эти упражнения на тестовых машинах только. Предполагается, что вы выполняете эти упражнения на виртуальных машинах на основе **KVM**. Для этой главы также предполагается, что вы можете изменить конфигурацию физической системы хоста для такой виртуальной машины.

Red Hat представляет свои экзамены в электронном виде. По этой причине лаборатории в этой и будущих главах доступны из средств массовой информации, которая сопровождает книгу. Лабораторные работы для этой главы находятся в подкаталоге глава 13. Если вы еще не настроили RHEL 7 в системе, обратитесь к главе 1 для установки инструкции.

Ответы для каждой лаборатории следуют за ответами самопроверки для вопросов, которые заполняются.

Лабораторная работа 1

В этой лабораторной работе вы настроите кэширующий сервер DNS-имен в локальной сети. В системе **DNS** должен быть настроен брандмауэр.

Лабораторная работа 2

В этой лабораторной работе вы настроите кэширующий **DNS-сервер**, который также перенаправляет запросы на один конкретный альтернативный **DNS-сервер**. Этот **второй DNS-сервер** может быть маршрутизатором домашней сети, **DNS-сервером**, назначенным интернет-провайдером, **DNS-сервером** для корпоративной сети или (для целей тестирования) **DNS-сервером** в файле **/var/named/named.ca**. Перед активацией сервиса обязательно очистите текущий кеш. Кроме того, в системе **DNS** должен быть настроен брандмауэр, ограничивающий доступ к локальной сети.

Лабораторная работа 3

Сконфигурируйте **Postfix** для включения аутентификации пользователя на физической хост-системе. Не вносите никаких дополнительных изменений в локальную систему. Убедитесь, что электронные письма могут быть отправлены от пользователя к пользователю локально. Пересылать электронные письма, направленные на локальную корневую учетную запись, на обычную учетную запись в локальной системе.

Лабораторная работа 4

Сконфигурируйте систему **Postfix** в Lab 3 для поддержки доступа из сети **example.com** или соответствующей **сети IP-адресов**. Сконфигурируйте систему на **server1.example.com** в качестве промежуточного узла **Postfix**, перенаправив его на систему физического узла.

Лабораторная работа 5

Настройте систему **Postfix в Lab 4** как нулевой клиент.

Лабораторная работа 6

Создайте логический том объемом **1 ГБ** на сервере **server1.example.com**. Используйте это блочное устройство в качестве резервного хранилища для нового **LUN на цели iSCSI с IQN iqn.2015-01.com.example: server1-disk1**. Настройте **tester1.example.com** в качестве **инициатора iSCSI с IQN iqn.2015-01.com.example: tester1**. Установите **ACL** для **цели iSCSI**, чтобы предоставить доступ только к **tester1**. Найдите удаленный **LUN на tester1** и создайте раздел с файловой системой **XFS**. Убедитесь, что том смонтирован при загрузке.

Лабораторная работа 7

В этой лабораторной работе вы создадите один **NTP-сервер** как одноранговый для второго обычного **NTP-сервера**.

ОТВЕТЫ НА САМОПРОВЕРКУ

Введение в службы доменных имен

1. Пакеты программного обеспечения **BIND** и **Unbound** предоставляют услуги DNS на RHEL 7.

Минимальные конфигурации DNS-сервера

2. Для поддержки доступа других клиентов к локальному **DNS-серверу**, убедитесь, что трафик TCP и UDP поддерживается через брандмауэр через порт 53 путем включения службы DNS на требуемой зоне **farewalld**.
3. Файл **/etc/named.conf** по умолчанию содержит базовый шаблон для сервера кэширования имен **DNS**.

4. Команда, обеспечивающая запуск службы **BIND DNS** при следующей загрузке **Linux**,

 # systemctl enable named

Разнообразие агентов электронной почты

5. На **RHEL 7** поддерживаются два примера **MTA**: **Postfix и sendmail**.
6. Командой, которая может помочь переключаться между **MTA Postfix и sendmail**, является

alternatives --config mta

Конфигурация Postfix

7. Самое простое решение - изменить директиву на

inet_interfaces = all

8. Адреса пересылки электронной почты обычно хранятся в **/etc/aliases**. Обязательно обработайте эти файлы в соответствующие базы данных; для **/etc/aliases** база данных обновляется командой **newaliases**.
9. Директива в файле **main.cf**, которая используется для указания домена, обслуживаемого сервером **Postfix**, имеет вид **mydestination**.

Цели и инициаторы iSCSI

10. Целевая служба должна быть запущена при перезагрузке на правильно настроенной **цели iSCSI**.
11. Командная оболочка **targetcli** может использоваться для настройки **цели iSCSI**.

Служба сетевого времени

12. Одна директива в файле **/etc/ntp.conf**, которая ограничивает доступ на основе отмеченных условий:

restrict 192.168.122.0 mask 255.255.255.0

ОТВЕТЫ ЛАБОРАТОРНОЙ РАБОТЫ

Лабораторная работа 1

В этой лабораторной работе вы можете использовать существующую конфигурацию в конфигурации **/etc/named.conf**. Все что вам нужно выполните следующие действия:

1. Установите **RPM-пакет bind**.
2. Измените директиву порта прослушивания 53 для включения локального IP-адреса; например, если местный **IP-адрес 192.168.122.150**, директива будет выглядеть так:

listen-on port 53 { 127.0.0.1; 192.168.122.150; };

3. Измените директиву **allow-query** для включения адреса локальной IP-сети:

allow-query {localhost; 192.168.122.0/24; };

4. Сохраните ваши изменения в **/etc/named.conf.**
5. Запустите названную службу:

 # systemctl start named

6. Измените локальный клиент так, чтобы он указывал на локальный сервер DNS-имен кэширования; заменить директиву **nameserver** в **/etc/resolv.conf** с **IP-адресом локальной**

системы. Например, если местный компьютер на **192.168.122.150**, директива будет иметь значение:

nameserver 192.168.122.50

7. Протестируйте новый локальный **DNS-сервер**. Попробуйте такие команды, как **dig www.mheducation.com**.
8. Укажите клиентские системы на **DNS-сервере**. Добавьте вышеупомянутую директиву **nameserver** к **/etc/resolv.conf** в этих удаленных клиентских системах:

   nameserver 192.168.122.50

9. Чтобы убедиться, что служба DNS запускается при следующей загрузке Linux, выполните следующую команду:

   # systemctl enable named
10. Откройте порты **TCP и UDP 53** в брандмауэре в локальной системе. Самый простой способ с помощью утилиты **firewall-cmd**:

    # firewall-cmd --permanent --add-service = dns
    # firewall-cmd --reload

Лабораторная работа 2

Как и в лабораторной работе 1, основное внимание в этой лабораторной работе уделяется настройке файла **/etc/named.conf**. Номинально **DNS-сервер** по умолчанию только для кэширования уже включает функции пересылки. Тем не менее, чтобы установить конкретный сервер пересылки, вы должны добавить запись пересылки, такую как следующая в разделе параметров:

forwarders { 192.168.122.1; };

Что касается других требований лаборатории, вы можете очистить текущий кеш с помощью команды **rndc flush** и перезагрузите файл конфигурации с помощью команды **rndc reload**.

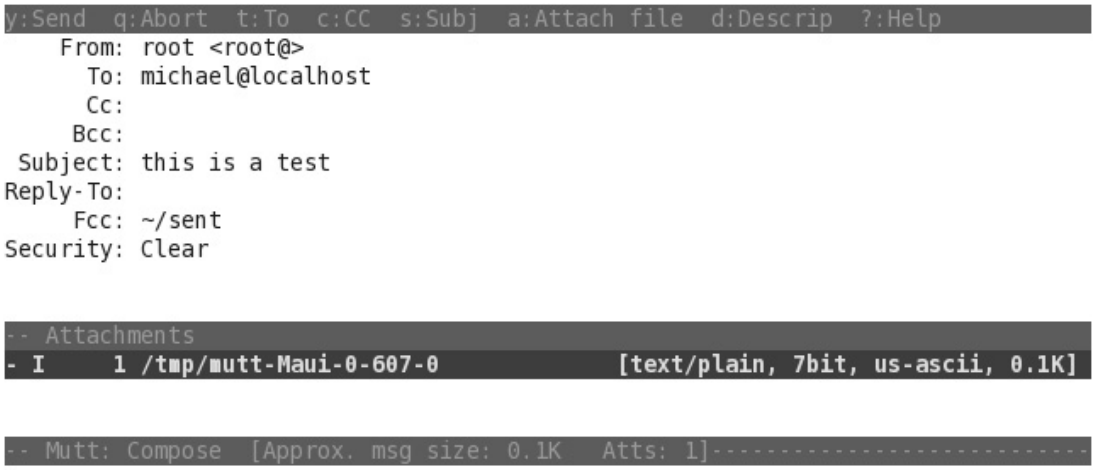
Лабораторная работа 3

Для лабораторий 3, 4 и 5 вы можете использовать почтовый клиент, такой как **Mutt**. Чтобы отправить электронное письмо пользователю **michael@localhost**, выполните следующие действия:

1. Запустите команду **mutt michael@localhost**. Должно появиться сообщение **To: michael@localhost.**
2. Нажмите ввод. В ответ на приглашение **Subject**: введите имя подходящего субъекта теста и нажмите **Enter**.
3. Вы оказались на пустом экране в редакторе vi. Используйте команды, соответствующие этому редактору в экран, аналогичный показанному на **рисунке 13-11**.
4. На экране, показанном на **рисунке 13-11**, нажмите **y**, чтобы отправить отмеченное сообщение.

Кроме того, вы можете проверять получение электронной почты в файле с именем пользователя в каталоге **/var/spool/mail**.
Обычно такую электронную почту можно просмотреть из учетной записи пользователя с помощью команды **mail** или **mutt**.

РИСУНОК 13-11 клиент Электронной почты Mutt



В **Postfix**, чтобы отключить локальный доступ в файле **/etc/postfix/main.cf**, измените директиву **inet_interfaces** принять **all** соединения:

inet_interfaces = all

Однако, чтобы соответствовать требованиям лаборатории, вы хотите сохранить значение по умолчанию этой директивы:

inet_interfaces = localhost

Как правило, для проверки подлинности на **SMTP-сервере** подключитесь из локальной системы с помощью команды **telnet localhost 25**. Когда вы видите сообщение, похожее на

220 maui.example.com ESMTP Postfix

введите следующую команду:

EHLO localhost

Чтобы подтвердить получение электронной почты в учетной записи пользователя, войдите в эту учетную запись или, по крайней мере, проверьте отметку времени связано с именем пользователя в каталоге **/var/mail**. Чтобы убедиться, что электронная почта, направленная пользователю **root**, перенаправленный на учетную запись обычного пользователя, вы добавите строку, подобную следующей, в файл **/etc/aliases**:

root: michael

Учитывая формулировку вопроса, любая стандартная учетная запись пользователя будет приемлемой. Конечно, чтобы реализовать это изменение, вам нужно будет запустить команду **newaliases**, которая обработает содержимое этого файла в файл **/etc/aliases.db**.

Лабораторная работа 4

Чтобы разрешить доступ не только с локального хоста, вам нужно изменить директиву **inet_interfaces** в **/etc/postfix/main.cf** на значение

inet_interfaces = all

Следующая задача - ограничить доступ к определенной сети (в данном случае **example.com**). Пока есть параметры в файлах **/etc/postfix**, возможно, наиболее эффективный способ ограничения доступа к определенной сети правило **firewalld rich rule**. Например, следующее пользовательское правило ограничит доступ к **TCP порт 25** для систем в данной сети IP-адресов. Показанная сеть основана на первоначально определенной конфигурация для **example.com**, сети **192.168.122.0/24**:

 # firewall-cmd --permanent --add-rich-rule='rule family=ipv4 source address=192.168.122.0/24 service \ name=smtp accept'

Кроме того, вы можете настроить эту сеть в файле **/etc/postfix/access** с помощью следующего правила:

192.168.122 OK

После запуска **Postfix** вы сможете подтвердить результат с помощью соответствующей команды **telnet** из удаленной системы. Например, если **Postfix** настроен в системе **с IP-адресом 192.168.122.50**, команда будет

 # telnet 192.168.122.50 25

Конфигурация промежуточного узла в **Postfix** основана на директиве **relayhost**. Для параметров в лаборатории, если физический хост находится в системе **maui.example.com**, директива в файле **main.cf** будет:

relayhost = maui.example.com

Если **Postfix** в системе **server1.example.com** правильно настроен как промежуточный узел, отправляйте электронные письма на адрес переадресованный хост должен быть надежно доставлен и авторизован в соответствующем файле **/var/log/maillog**.

Лабораторная работа 5

Конфигурация **Postfix** как нулевого клиента проста и представлена в **таблице 13-6**.
Как минимум, вы должны настроить **myorigin, mydestination, local_transport** и **relayhost** директивы. Другие директивы, такие как **myhostname, mydomain** и **mynetworks**, уже должны иметь соответствующие значения по умолчанию.

Подтвердите настройки с помощью команды **postconf -n** и запустите службу. Не забудьте настроить **Postfix** для автоматического запуска при следующем включении машины.

Лабораторная работа 6

Это длительная лабораторная работа, но она очень похожа на пример конфигурации, описанный в разделе «**Цели и задачи iSCSI. Инициаторы**», в разделе этой главы. Пожалуйста, обратитесь к этому разделу для углубленного обсуждения.

Лабораторная работа 7

В этой лабораторной работе вы настроите один **NTP-сервер** как одноранговый для другого. Это возможно с помощью директивы **peer**, настраивается в файле конфигурации **/etc/ntp.conf**. Например, если настроен обычный **NTP-сервер** на **IP-адресе 192.168.122.50** вы можете настроить одноранговый узел на сервере **192.168.122.150** с помощью следующей директивы:

peer 192.168.122.50

Просто запомните, одноранговый узел **NTP** не работает, если параметр **nopeer** не был удален из директива **restrict** в файле **ntp.conf.**