Расширенное администрирование Linux

Блок 3

v 1.04

Оглавление

DNS

С увеличением количества компьютеров в сети стало неудобно обращаться к ним по IP адресу, поэтому компьютерам начали присваивать имена. Первоначально для преобразования имени компьютера в IP адрес и обратно использовался файл /etc/hosts. Одним из условий правильного преобразования имен при помощи этого файла является наличие его на всех компьютерах в сети, причем с одинаковым содержанием.

Когда в сети несколько десятков компьютеров, синхронизация файла hosts между ними не вызывает особых проблем. Но если представить себе то количество машин, которое сейчас присутствует в Internet, то сразу становится понятно, что использование файла hosts вызовет очень большие затруднения. Представьте себе, что при подключении новой машины в сеть необходимо обновить этот файл на всех машинах сети. Кроме того, следует учитывать и размер этого файла, в котором на каждый компьютер отводиться одна строка. То есть, если в качестве источника данных для преобразования использовать /etc/hosts, то весь трафик в сети будет использоваться только для синхронизации этого файла.

В начале 80-х годов была создана система DNS — Domain Name System. Эта система представляет из себя распределенную базу данных, предназначенную для преобразования имен компьютеров в IP адреса и наоборот. DNS лишена недостатков файла hosts и позволяет довольно быстро осуществлять преобразование имен.

Пространство имен DNS

В DNS было введено понятие домена. Любая машина в сети находится в определенном домене DNS.

Вы не должны путать домены DNS и домены Windows. Они используются для разных целей. Домены DNS применяются только для преобразования имен. Домены Windows связаны с вопросами безопасности.

Доменная структура DNS является иерархической. Иерархия имен начинается с корневого домена, обозначаемого символом точка — «.». В корневом домене находятся домены первого уровня. Они могут быть организованы по географическому принципу: ru, ua, fi и т.д. И по историческому принципу: домены, которые изначально использовались в США: com, net, mil и т.д. Дальше идут домены второго, третьего и т.д. уровней.

В DNS существует понятие полностью квалифицированного доменного имени — FQDN. Такое имя состоит из двух частей: имени компьютера и имени домена.

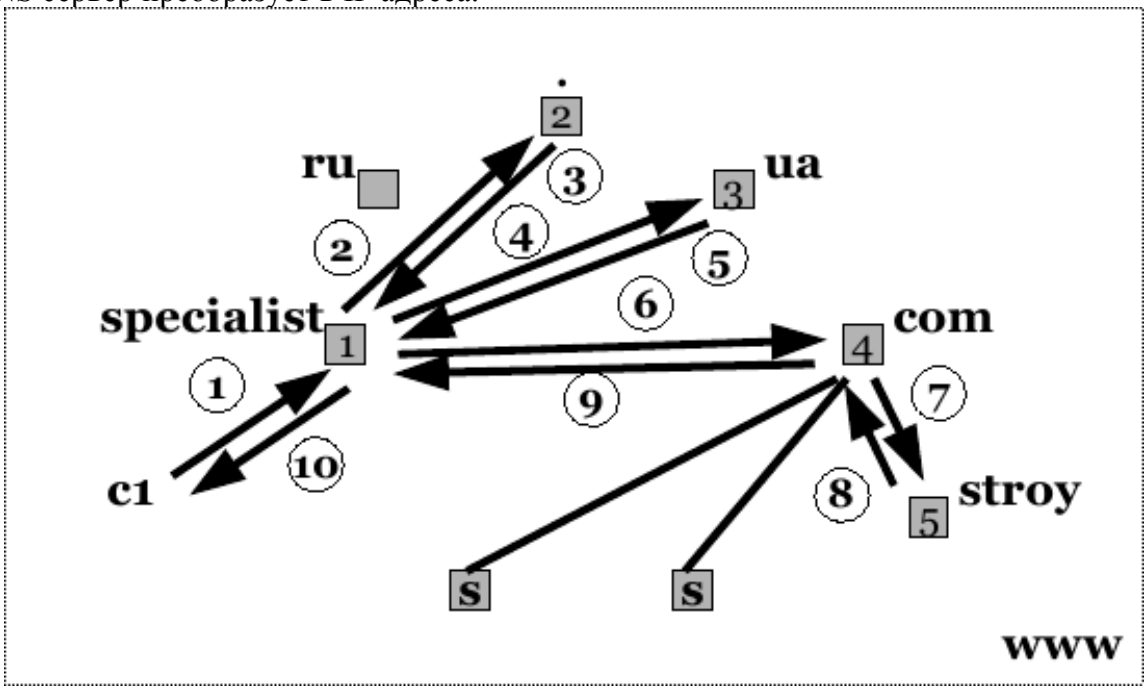
Например:

- www.specialist.ru – состоит из имени машины www и имени домена — specialist.ru
- c1.of.specialist.ru – имя машины c1 и имя домена of.specialist.ru

Как видно из примеров, первое имя — это имя машины. Оставшаяся часть — это домен.

Зона ответственности DNS сервера

Если предположить, что каждый DNS сервер в Интернет отвечает за все компьютеры в сети, то это не ничем не отличается от использования файла hosts. Поэтому DNS выполнена в виде распределенной базы данных. Каждый DNS сервер имеет свою зону ответственности. Он не отвечает за весь Интернет, а только за небольшую часть его. Зона ответственности — это компьютеры в определенном домене, имена которых DNS сервер преобразует в IP адреса.



Например, DNS сервер, отвечающий за домен specialist.ru предназначен для преобразования имен компьютеров, входящих в этот домен. Если посмотреть на схему, то становиться понятно, что этот DNS сервер может преобразовать имена компьютеров: office.specialist.ru, www.specialist.ru, ftp.specialist.ru. Компьютеры c1.office.specialist.ru и c2.office.specialist.ru относятся к другому домену и поэтому их имена не будут преобразованы этим DNS сервером.

Принцип работы DNS

Для того, чтобы понять принцип работы DNS воспользуемся схемой, приведенной выше. Предположим, что все DNS сервера были включены одновременно и не отвечали ни на один запрос. Предположим, что DNS сервера, отвечающие за определенный домен (на схеме обозначены квадратом), находятся на компьютере с именем этого домена. В реальной жизни DNS сервера могут поддерживать сразу несколько доменов одновременно и располагаться в любом месте сети.
Работая на машине c1.specialist.ru, в браузере набираем www.stroy.com.ua и нажимаем Enter. Для того, чтобы браузер смог послать пакеты на соответствующий WEB сервер, ему необходимо получить IP адрес этого сервера. Он обратиться к клиенту DNS, находящемуся на том же компьютере, а клиент обратится к DNS серверу, на работу с которым он настроен.
Предположим, что DNS сервер, обслуживающий нашего клиента находится на машине specialist.ru. Именно на этот сервер пошлет первый запрос наш клиент (шаг 1 на схеме). Поскольку машина www.stroy.com.ua не входит в зону ответственности DNS сервера specialist.ru, этот сервер не может дать ответ на вопрос клиента и обратиться к DNS серверу, отвечающему за корневой домен (шаг 2 на схеме).
Корневой DNS сервер тоже не отвечает за машины домена stroy.com.ua и не может дать окончательный ответ. Но этот сервер знает IP адрес и имя DNS сервера, отвечающего за домен ua. И он возвращает эти данные нашему серверу (шаг 3 на схеме).

Следующий запрос будет послан DNS серверу, отвечающему за домен ua (шаг 4 на схеме). Но и он не может дать нам ответ на наш запрос, т.к. не является ответственным за домен stroy.com.ua. Но он знает IP адрес и имя DNS сервера, отвечающего за домен com.ua и возвращает нам эти значения (шаг 5 на схеме).

Наш DNS сервер посылает запрос на DNS сервер, отвечающий за домен com.ua (шаг 6 на схеме). Этот DNS сервер не является ответственным за домен stroy.com.ua, но ему известны IP адрес и имя DNS сервера, отвечающего за домен stroy.com.ua и он самостоятельно посылает запрос на этот сервер (шаг 7 на схеме). Почему он самостоятельно ищет информацию, будет рассказано ниже.

Поскольку DNS сервер отвечает за домен stroy.com.ua, он возвращает IP адрес машины www (шаг 8 на схеме). Этот IP адрес возвращается нашему DNS серверу (шаг 9) и он возвращает его клиенту (шаг 10).

Кэширующие DNS сервера

По статистике на одной WEB страничке в среднем находится около 15-ти картинок. И если предположить, что мы используем старый WEB браузер, не поддерживающий протокол HTTP версии 1.1, то на каждую картинку будет формироваться отдельный запрос. Произойдет около 15-ти запросов к нашему DNS серверу и вся процедура определения IP адреса будет повторяться при каждом запросе. Если бы все было именно так, как описано выше, то подавляющая часть интернет трафика состояла бы из DNS запросов.

Для того, чтобы не возникало проблем с повторяющимися запросами, все DNS сервера являются кэширующими. Наш сервер поместит информацию о машине www.stroy.com.ua в свой локальный кэш и при следующем запросе выдаст информацию из локального кэша. Кроме того он поместит в кэш ответы DNS серверов, обозначенных на схеме номерами 2, 3 и 4.

Время хранения информации в кэш сервера зависит от настроек зоны. В большинстве случаев оно составляет около суток.

Рекурсивные и не рекурсивные DNS сервера

Если посмотреть на приведенную схему, то можно обратить внимание на различие в поведении DNS серверов 2, 3 и 4. Первые два сервера выдавали информацию только о нижестоящих в иерархии DNS серверах. А сервер 4 сам обратился за информацией к серверу 5.

Первые два сервера являются не рекурсивными, а сервер 4 рекурсивным. Рекурсивные сервера пытаются самостоятельно выполнить все шаги по получению интересующей информации, не рекурсивные выдают только информацию о зоне, за которую они отвечают или информацию находящуюся в их кэш.

DNS сервера, отвечающие за домены первого и второго уровня, обычно являются не рекурсивными, поскольку им приходиться обрабатывать большое количество запросов. Например, DNS сервера, отвечающие за корневой домен, обрабатывают около 20000 запросов в секунду.

DNS сервер, обозначенный на схеме под номером 4, тоже можно сделать не рекурсивным. Будет ли ваш DNS сервер рекурсивным или не рекурсивным зависит от того, как вы его настроите.

DNS сервера, обслуживающие запросы клиентов DNS, обязательно должны быть рекурсивными для запросов клиентов, поскольку клиенты DNS не понимают пересылок к другим DNS серверам.

DNS сервер, который будет рассматриваться на нашем курсе, позволяет сделать так, что для запросов внутренних клиентов DNS он будет выглядеть как рекурсивный сервер, а для внешних запросов как не рекурсивный.

Авторитетные и не авторитетные ответы

Предположим, что DNS сервер, отвечающий за домен stroy.com.ua, по каким-либо причинам не доступен. Тогда DNS сервер 4 не сможет получить информацию о машине www.stroy.com.ua и по истечении некоторого времени, отведенного на запрос, вернет нам отрицательный ответ — машина www.stroy.com.ua не существует. Эта информация попадет в кэш нашего сервера и будет выдана клиенту.

Теперь предположим, что через некоторый промежуток времени, DNS сервер, отвечающий за домен stroy.com.ua, станет доступен и начнет отвечать на вопросы. Но поскольку информация уже храниться в кэше нашего сервера, в среднем в течении суток наш сервер будет выдавать отрицательные ответы клиентам.

Чтобы такой ситуации не возникало, рекомендуется использовать дополнительные DNS сервера, поддерживающие ваш домен. На схеме такие сервера обозначены символом S. Серверу 4 известны все DNS сервера, отвечающие за домен stroy.com.ua и, если сервер 5 не отвечает на запросы, сервер 4 обратиться к другим DNS серверам, ответственным за домен.

Сервер 5 называется главным (master) DNS сервером. Дополнительные сервера называются подчиненными (slave) DNS серверами. Все изменения в домене описываются только на master сервере, на slave серверах информация непосредственно не редактируется, они получают ее с master сервера.

Когда клиент DNS получает информацию с авторитетных серверов — master или slave, этот ответ называется авторитетным. Если информация получена из кэш, такой ответ называется не авторитетным.

Вопросы

1. К какому домену принадлежит машина c230.unix.specialist.ru?

2. Может ли один DNS сервер поддерживать несколько доменов одновременно?

3. Могут ли клиенты DNS работать с не рекурсивными DNS серверам?

4. Вы создаете DNS сервер предприятия, поддерживающий ваш домен и обслуживающий внутренних клиентов. Можно ли сделать так, чтобы он для внешних запросов был не рекурсивным, а для внутренних рекурсивным?

5. Обязательно ли наличие slave серверов, которые будут поддерживать вашу зону?

DNS сервер BIND

Сервер BIND (Berkeley Internet Name Domain) – наиболее популярный в UNIX DNS сервер. Он распространяется организацией ISC (Internet Software Consorortium) в исходных кодах. Поставляется со всеми дистрибутивами Linux, в том числе и с Ubuntu Server.

BIND был создан в 1985 году Кевином Данлапом (Kevin Dunlap). На данный момент его последняя версия – 9. Он поддерживает все основные особенности системы DNS, а также все новинки, которые были добавлены в последнее время, такие как:

- поддержка Ipv6
- DNSSEC
- TSIG
- поддержка расширенного протокола DNS — EDNS0

Особо хочется отметить поддержку EDNS0 сервером BIND. В расширенном протоколе добавлена возможность использования протокола TCP не только для передачи зон, но и для передачи запросов, в том числе и с цифровыми подписями. Кроме того, поддерживается инкрементальная пересылка зон и динамическое обновление зон (для DHCP серверов).

Включение поддержки домена

Конфигурационные файлы сервера BIND
Поднятие поддержки домена unix.specialist.ru на DNS сервере преподавателя

Дальнейшее изучение DNS сервера BIND будет происходить следующим образом:

1. Сначала преподаватель объяснит, как поднять поддержку домена unix.specialist.ru. Домен будет поднят на DNS сервере преподавателя. При этом будут показаны все конфигурационные файлы и все изменения, которые будут внесены в эти файлы.
2. Так же будут показаны программы для проверки синтаксиса конфигурационных файлов, запуск и работа с демоном DNS сервера — named.
3. Затем слушателям будут делегированы права на поддомены в домене unix.specialist.ru и на отдельной лабораторной работе они самостоятельно поднимут поддержку этих доменов на своих серверах.
4. После запуска DNS серверов на машинах слушателей будут рассказаны различные аспекты применения DNS сервера BIND в различных ситуациях.

Конфигурационные файлы сервера BIND

Основной конфигурационный файл — named.conf

Файл описания зоны

Основным конфигурационным файлом сервера BIND является файл /etc/bind/named.conf. Кроме файла named.conf, используются дополнительные конфигурационные файлы:

- описания зон
- описания ключей
- файлы подсказок

named.conf

В файле определяются основные конфигурационные параметры DNS сервера и зоны, которые он поддерживает.

Инструкции, применяемые в файле:

•include

•options

•acl

•zone

•view

и другие.

В файле named.conf определяются основные параметры сервера BIND, а так же зоны, которые он поддерживает. В файле можно использовать комментарии в стиле C/C++ и shell script:

```
/* Это комментарий, который можно располагать на нескольких строках */
// Комментарий на одной строке.
# Комментарий на одной строке.
```

Файл состоит из инструкций. Каждая инструкция начинается с ключевого слова, определяющего ее тип и должна завершаться символом «;». Если параметры инструкции не помещаются на одну строку, их необходимо брать в фигурные скобки: { и }.

Список поддерживаемых инструкций приведен в таблице:

Инструкция	***Описание***
include	Подключает внешний файл
options	Определяет глобальные параметры сервера BIND
server	Задает параметры сервера
lwres	Конфигурирует12 сервер BIND 9 в качестве упрощенного распознавателя
key	Определяет параметры аутентификации
acl	Определяет списки управления
zone	Определяет зоны, поддерживаемые сервером

trusted-keys	Определенные в конфигурационном файле ключи шифрования
controls	Определяет, как утилита rndc будет управлять сервером BIND
logging	Определяет категории журнальных сообщений и каналы их распространения
view	Определяет представление пространства имен

В инструкциях достаточно часто необходимо описывать IP адреса машин и сетей. Ниже приведены возможные варианты их записи:

- IP адрес. Например, 178.11.201.64
- Адрес сети с маской подсети. Например, 193.16.18/24
- Имя ранее определенного списка контроля доступа (acl)
- Оператор отрицания — «!»

Инструкция options

При помощи инструкции options задаются глобальные параметры сервера BIND.

```
options {
        параметр;
        параметр;
};
```

Ниже приведены некоторые параметры, определяемые в инструкции options:

Параметр	***Описание***
directory	Определяет директорию, в которой сервер BIND будет искать файлы описания зон и создавать различные дополнительные файлы. Обычно этот параметр ссылается на директорию /var/named. Пример: **`directory «/var/named»`**
notify yes|no	Если параметр notify равен yes, то при изменении описания зоны будут посланы уведомления всем slave DNS серверам. Значение по умолчанию — yes. Пример: **`notify no`**
also-notify address	При помощи этого параметра определяются DNS сервера, которые необходимо уведомить при изменении зоны. Slave DNS сервера в этом списке указывать не надо. Значение по умолчанию не определено. Пример: **`also-notify { 193.12.20.1; 193.12.38.200; };`**
recursion yes|no	Параметр определяет, будет ли сервер BIND рекурсивным сервером. Значение по умолчанию — yes. Пример: **`recursion yes`**

allow-recursion address	Этот параметр определяет адреса машин или сетей, для которых сервер BIND будет выступать в роли рекурсивного сервера. Значение по умолчанию не определенно. Пример: **allow-recursion { 193.12.13.240; 194.12.34/24; };**
listen-on port порт list	Параметр позволяет определить, на каком интерфейсе и порту будет слушать запросы сервер. Значение по умолчанию: все интерфейсы, порт 53. Пример: **listen-on { 5.6.7.8; };** **listen-on port 1234 { ! 1.2.3.4; 1.2/16; };**
allow-query адреса	Параметр определяет, с каких адресов можно посылать запросы нашему серверу. Значение по умолчанию: разрешены все адреса. Пример: **allow-query { 1.2.3.4; 10.10.100/24; };**
allow-transfer адреса	Параметр определяет, на какие сервера разрешены зонные пересылки. Значение по умолчанию: пересылки разрешены всем серверам. Пример: **allow-transfer { 1.2.3.4; 10.10.100/24; };**
blackhole адреса	Параметр определяет адреса серверов, запросы от которых будут всегда игнорироваться. Сервер не будет посылать им свои запросы. Значение по умолчанию: список не определен. Пример: **blackhole { 1.2.3.4; 2.3.4.5; };**

Инструкция acl

ACL — список управления доступом. При помощи инструкции acl можно определить список часто используемых IP адресов и присвоить ему имя, на которое в дальнейшем будут ссылки в параметрах других инструкций.

```
acl имя {
        элементы списка;
};
```

Определение acl должно происходить до того, как его имя будет использоваться в параметрах.

Существуют четыре заранее определенных списка:

- any – соответствует всем узлам
- localnets – соответствует всем узлам локальной сети
- localhost – соответствует своему компьютеру
- none – не соответствует ни одному узлу

Пример определения acl:

```
acl internal { 1.2.3.4; 2.3.4.5; 10.10.100/24; 192.168.0/24; };
```

Инструкция zone

Инструкция предназначена для описания зон ответственности DNS сервера. Для каждого поддерживаемого сервером DNS домена необходимо писать отдельную инструкцию zone.

Параметры, используемые в инструкции, зависят от типа зоны. Тип зоны определяется при помощи параметра type.

Пример описания зоны:

```
zone «имя домена» IN {
        type тип_зоны;
        параметры зоны;
};
```

В таблице перечислены возможные типы зон.

Тип	***Описание***
master	Определяет master зону
slave	Определяет slave зону
hint	Определяет зону подсказку
forward	Определяет зону переадресации

Все вышеперечисленные типы зон будут рассмотрены ниже при описании поднятия поддержки домена и при рассмотрении различных случаев применения DNS сервера BIND.

Файл описания зоны

Специальные символы

Директивы

Записи:

•SOA

•NS

•MX

•A

•CNAME

•PTR

и другие

Основная информация о конкретном домене располагается в файле описания зоны. В нем описываются так называемые «записи о ресурсах», которые определяются в RFC: 882, 1035, 1183, 2065, 2181, 2308 и 2535.

В файле описания зоны можно использовать следующие специальные символы:

; — комментарий.

@ — Имя текущего домена. Берется либо из инструкции zone, либо определяется в файле описания зоны директивой $ORIGIN.

() — Разбивка данных на несколько строк.

***** — используется только в именах машин или доменов.

Директивы

В файле описания зоны можно использовать специальные директивы:

$TTL время — определяет время жизни записей в кэш DNS сервера для всех записей зоны

$ORIGIN домен — определяет имя домена, подставляемое вместо символа @, или подставляется в конце не полностью определенного имени компьютера или домена

$INCLUDE файл— подключает внешние файлы

$GENERATE параметры — предназначена для создания наборов похожих записей

Особенности написания FQDN имен в файлах описания зоны

В файлах описания зоны. только в файлах описания зоны! При указании полностью квалифицированного доменного имени машины (FQDN) или полного имени домена. Требуется явно указывать имя корневого домена — «.».

Например, если необходимо указать FQDN имя машины master.unix.specialist.ru в файле описания зоны, его необходимо описывать следующим образом:

```
master.unix.specialist.ru.
```

Обратите внимание на точку в конце имени — это явное указание корневого домена.

Если в конце имени точки нет, то DNS сервер автоматически подставит имя текущего домена (имя берется либо из инструкции zone, конфигурационного файла named.conf, либо определяется при помощи директивы $ORIGIN).

Например, если текущий домен unix.specialist.ru. И если в файле описания зоны его имя написано без точки — master.unix.specialist.ru, то в результате DNS сервер подставит следующее значение:

```
master.unix.specialist.ru.unix.specialist.ru.
```

Наиболее распространенная ошибка при создании файла описания зоны — это отсутствие точки в конце FQDN.

Записи о ресурсах

Формат записи можно представить следующим образом:

```
[имя_компьютера|имя_домена] [ttl] [класс] тип параметры
```

Первое – это имя машины или домена. Что именно писать в этом поле, зависит от типа записи. Если это поле оставить пустым, то его значение берется из предыдущей записи.

Если вы не указываете первое поле, то в начале строки обязательно должен присутствовать либо символ пробела, либо табуляции.

Второе поле – время жизни записи в кэш DNS сервера. Значение поля устанавливается в секундах. Если поле не определено, его значение берется из значения по умолчанию для данной зоны.

Третье поле – класс сети. Можно использовать следующие классы сетей:

- IN – Internet (значение по умолчанию
- CH – ChaosNet. В настоящее время не используется.
- HS – Hesoid – информационная служба, являющаяся надстройкой пакета BIND. Используется крайне редко

Тип записи – это зарезервированное слово. Основные типы записей приведены в

таблице.

Тип	*Описание*
SOA	Определение параметров зоны DNS. Обязательная запись
NS	Определение DNS серверов, авторитетных для зоны. Делегирование полномочий поддоменам. Обязательная запись
A	Преобразование имени в IP адрес
AAAA	Преобразование имени в адрес IPv6
A6	Преобразование имени в адрес IPv6
PTR	Преобразование IP адреса в имя
MX	Применяется для указание почтового сервера, отвечающего за почту домена
KEY	Открытый ключ шифрования для DNS имени
CNAME	Дополнительное имя машины (псевдоним)
SRV	Определение служб в пределах домена

В любом файле описания зоны должны быть определены три обязательные записи: SOA, NS и A для NS (или PTR в случае зоны обратного преобразования).

Запись SOA

Запись SOA (Start Of Autority) – определяет начало описания зоны. Это одна из обязательных записей, которая должна присутствовать в файле описания зоны. Она должна быть самой первой в файле описания зоны. Описание зоны в файле продолжается до тех пор, пока не встретиться другая запись SOA.

Пример формата записи SOA:

```
unix.specialist.ru. IN SOA master.unix.specialist.ru. (
        artur.unix.specialist.ru. ; e-mail
        2008072501 ; серийный номер записи
        18H ; время обновления
        20M ; интервал между попытками
        2W ; интервал устаревания
        1D ) ; TTL
```

Рассмотрим из каких полей состоит запись SOA.

- Первый параметр – это имя домена. В этом параметре можно использовать специальный символ @, вместо которого будет подставлено имя текущего домена.
- Второй (необязательный) параметр TTL не указан, поэтому время жизни этой записи в кэш DNS сервера будет взято из значения по умолчанию для данной зоны.
- Третий параметр – класс сети определен как IN. Несмотря на то, что это параметр не обязательный, его рекомендуют определять для улучшения читабельности файлов описания зоны.

Дальше идет тип записи SOA. Все остальные параметры – это параметры присущие только записи типа SOA.

- Сначала указывается имя master DNS сервера, ответственного за данную зону
- Второй параметр — почтовый адрес человека, отвечающего за данную зону. Адрес может находиться в любом домене.
- 2008072501 – серийный номер записи зоны. При изменении файла описания зоны

вы обязаны увеличить это число, потому что оно используется slave DNS серверами для определения необходимости зонной пересылки. Если серийный номер на slave сервере больше или равен серийному номеру на master DNS сервере, зонной пересылки не будет. Если он меньше, то будет происходить зонная пересылка на slave сервер.

  - В качестве номера можно использовать: 1, 2, 3, 4 и т.д. Но в реальной практике рекомендуется в серийном номере указывать дату последнего изменения файла описания зоны. В приведенном примере, последнее изменение в зоне было в 2008 году, в июле месяце, 25 числа. В этот день было одно изменение, о чем свидетельствует число 01.
  - Максимальное количество знаков, которое можно применять в этом поле, равняется десяти.

- Дальше идет число, определяющее интервал времени, через который slave сервер будет обращаться к master серверу для сравнения серийных номеров записей.
- В параметрах записи типа SOA интервал времени указывается в виде количества секунд. Но сервер BIND позволяет использовать сокращенный вариант определения времени.
- Если slave сервер не может подключиться к master серверу, он переходит в другой режим работы, при котором интервал времени обращения к master серверу уменьшается. Параметр 20M определяет этот интервал.
- Slave сервер не может до бесконечности пытаться подключиться к master серверу, поэтому следующий параметр 2W определяет время, по прошествии которого slave сервер перестает поддерживать эту зону.
- Последний параметр определяет время жизни по умолчанию в кэш DNS серверов отрицательных ответов нашего сервера.

Все параметры записи типа SOA являются обязательными.

Запись NS

Запись типа NS (Name Server) предназначена для описания всех DNS серверов, авторитетных для данного домена. При помощи этой записи вы должны описать все master и slave сервера. Она так же применяется при делегировании прав на зону.

Запись NS относиться к обязательным записям. В файле описания зоны должна быть определена как минимум одна такая запись. Формат записи NS:

```
[зона] [TTL] [IN] NS имя_сервера
```

Если предположить, что за зону unix.specialist.ru отвечают DNS сервера: master.unix.specialist.ru. и ns.example.ru., в файл описания зоны необходимо добавить две записи:

```
@ IN NS master.unix.specialist.ru.
@ IN NS ns.example.ru.
```

Кто из перечисленных серверов является master, а кто slave определяется при помощи первого параметра записи SOA.

Записи MX

Записи типа MX (Mail Exchanger) предназначены для указания почтовых серверов, отвечающих за прием почты для домена. Формат записи MX:

```
[домен] [TTL] [IN] MX приоритет почтовый_сервер
```

При создании почтовой системы можно точно так же, как и в DNS, выделить основной и вспомогательный почтовый сервера. Приоритет отправки почты задается при помощи поля «приоритет». Чем меньше число в этом поле, тем выше приоритет

указанного сервера. То есть, по умолчанию почта будет отправляться на почтовый сервер с большим приоритетом. Но если он, по каким либо причинам не будет доступен, почта будет отправляться на сервер с меньшим приоритетом. Почтовые сервера должны быть сконфигурированы соответствующим образом.

Предположим, что почту для домена unix.specialist.ru могут принимать два почтовых сервера: smpt.unix.specialist.ru и smtp.example.ru. Тогда в файле описания зоны необходимо добавить две записи:

```
@ IN MX 5 smtp.unix.specialist.ru.
@ IN MX 10 smtp.example.ru.
```

Запись MX не обязательная.

Запись A

Запись типа A (address) предназначена для преобразования имени машины в IP адрес. Формат записи:

```
[имя_машины] [TTL] [IN] A IP_адрес
```

Предположим, что в домене unix.specialist.ru есть три машины: master.unix.specialist.ru, c1.unix.specialist.ru и c2.unix.specialist.ru. Тогда в файле описания зоны необходимо добавить три записи:

```
master IN A 10.10.108.20
c1     IN A 10.10.108.1
c2     IN A 10.10.108.2
```

IP адреса машин могут находиться в разных сетях. Если машина имеет несколько сетевых интерфейсов, возникает желание написать две записи для одной машины. Но если это сделать, вас ждет сюрприз. Например, машина c1.unix.specialist.ru имеет два сетевых интерфейса со следующими IP адресами: 10.10.108.1 и 192.168.0.1.

Если в файл описания зоны поместить следующие строки:

```
c1     IN A 10.10.100.1
       IN A 192.168.0.1
```

Тогда при первом запросе на преобразование имени машины c1.unix.specialist.ru будет выдан первый IP адрес. При втором — второй. При третьем — снова первый. При четвертом — опять второй и т.д. Поэтому рекомендуется в DNS сервере одной машине присваивать один IP адрес.

Иногда эту особенность используют для распределения нагрузки. Например, у вас есть два WEB сервера, обслуживающие WEB сайт. Если в файле описания зоны машине www определить две записи A, то произойдет распределение запросов на подключение между этими WEB серверами. Правда, следует учитывать, что запись попадет в кэш DNS сервера и клиенты, использующие этот сервер в течении суток, будут попадать только на один из WEB серверов.

Запись CNAME

Запись типа CNAME (Canonical Name) — позволяет добавить несколько имен одной и той же машине. Формат записи CNAME:

```
дополнительное_имя [TTL] [IN] CNAME каноническое_имя
```

Одной машине можно присвоить несколько имен, в том числе и в разных доменах, которые преобразуются в один IP адрес. Обратное преобразование IP адреса возможно только в одно имя машины. Такое имя называется каноническим. При помощи записи типа A определяют только канонические имена. Все дополнительные имена необходимо определять только при помощи записи типа CNAME.

Предположим, что машине master.unix.specialist.ru необходимо присвоить дополнительные имена: www.unix.specialist.ru и ftp.unix.specialist.ru. Тогда в файл описания зоны будут добавлены следующие строки:

```
www IN CNAME master
ftp IN CNAME master
```

Важно запомнить, что параметр записи CNAME — это имя машины, а не ее IP адрес.

Запись PTR

Записи типа PTR (Pointer) предназначены для обратного преобразования — IP адреса в имя машины. Обычно эти записи применяются в зонах обратного преобразования. Формат записи:

```
адрес [TTL] [IN] PTR имя
```

Более подробно этот тип записи будет рассмотрен при поднятии зоны обратного преобразования.

Включение поддержки домена unix.specialist.ru на DNS сервере преподавателя

Параметры файла named.conf

Файл описания зоны

В этом разделе будут показаны все действия, которые необходимо выполнить при поднятии поддержки домена в DNS сервере BIND.

Параметры файла named.conf

```
zone “unix.specialist.ru” IN {
     type master;
     file “master.unix.specialist.ru”;
};
```

На DNS сервере преподавателя необходимо поднять поддержку домена unix.specialist.ru. Первое, что следует сделать – в конфигурационном файле named.conf при помощи инструкции zone описать домен. Для этого в файл добавляем такие строки:

```
zone «unix.specialist.ru» IN {
       type master;
       file «master.unix.specialist.ru»;
};
```

В примере перечислены только необходимые параметры инструкции zone:

type master – определяет, что указанная зона будет master зоной.

file «master.unix.specialist.ru» — определяет имя файла, в котором будут описаны параметры зоны и компьютеры домена. Сервер BIND будет искать этот файл в директории, определяемой параметром directory в инструкции options. Имя файла может быть любым, но если предполагается, что ваш DNS сервер будет поддерживать несколько master и slave зон, рекомендуется имена файлов описания master зон начинать со слова master, а slave зон – со слова slave.

Остальные параметры, которые можно использовать при описании master зоны, будут рассмотрены в других разделах.

Файл описания зоны

```
$TTL 60
@  IN SOA ...
   IN NS ...
   IN MX ...
c1 IN A ...
www IN CNAME ...
```

Для описания зоны unix.specialist.ru создаем файл описания зоны /var/named/master.unix.specialist.ru. В первую очередь определяем время жизни по умолчанию для всех записей зоны:

```
$TTL 60
```

В реальных условиях значение должно быть равно одним суткам — 1D или близкое к нему значение. Но поскольку в классе в записи зон регулярно будут вноситься изменения, TTL равняется 60ти секундам.

Создание записи типа SOA

Самой первой записью в файле описания зоны обязательно должна быть запись типа SOA.

```
unix.specialist.ru. IN SOA master.unix.specialist.ru. (
        root.unix.specialist.ru.
        2008072501
        18H
        20M
        2W
        60 )
```

Если использовать особенности написания имен и применение специальных символов в файле описания зоны, первую строку можно сократить.

- Во-первых, вместо имени домена поставить символ @
- Во-вторых, в нашем случае master DNS сервер находится в домене, который он поддерживает, поэтому имя DNS сервера можно написать сокращенно — master, без точки в конце. Имя домена сервер подставит сам

То же самое касается почтового адреса человека, ответственного за зону. В этом поле можно написать root, без точки в конце. Имя домена сервер подставит автоматически.

В результате запись SOA будет выглядеть следующим образом:

```
@ IN SOA master root (
        2008072501
        18H
        20M
        2W
        60 )
```

Последнее поле – время жизни в кэш DNS сервера отрицательных ответов, как и глобальная директива $TTL, должно быть равным одному дню. Но поскольку в классе записи будут меняться достаточно часто, этот параметр устанавливается равным 60-ти секундам.

Определение DNS серверов, ответственных за домен

Поскольку в условиях нашего класса за домен unix.specialist.ru отвечает только один DNS сервер, в файл описания зоны будет добавлена только одна запись типа NS.

```
@ IN NS master.unix.specialist.ru.
```

Как и в записи SOA, можно сократить запись NS. Первое поле оставить пустым. Его значение будет взято из предыдущей записи SOA. Можно сократить имя DNS сервера, отвечающего за зону. В результате запись будет выглядеть следующим образом:

```
IN NS master
```

Вместо пустого первого поля необходимо подставлять символ пробел или табуляция.

Определение почтовых серверов, принимающих почту для домена

Почту для домена unix.specialist.ru будет принимать только один почтовый сервер — master.unix.specialist.ru. Поэтому в файл описания зоны будет добавлена одна запись типа MX.

```
IN MX 5 master
```

Запись типа MX идет сразу за записью NS, поэтому первое поле можно не указывать, его значение будет взято из предыдущей записи.

Описание компьютеров, входящих в домен

Все компьютеры, входящие в домен unix.specialist.ru, необходимо описать при помощи записей типа A.

```
master IN A 10.10.108.20
c1     IN A 10.10.108.1
c2     IN A 10.10.108.2
```

И так далее.

Описание псевдонимов машин

Все псевдонимы описываются при помощи записи типа CNAME.

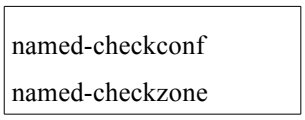
```
www IN CNAME master
ftp IN CNAME master
pop IN CNAME master
```

Обратите внимание на то, что имена машин DNS серверов и почтовых серверов, определяемых при помощи записей NS и MX, должны быть каноническими. В этих записях не рекомендуется использовать псевдонимы, определяемые при помощи записи CNAME.

В результате файл описания зоны будет выглядеть следующим образом:

```
$TTL 60
@      IN SOA master root (
              2008072502
              18H
              20M
              2W
              60 )
       IN NS master
       IN MX 5 master
master IN A 10.10.108.20
c1     IN A 10.10.108.1
c2     IN A 10.10.108.2
c3     IN A 10.10.108.3
c4     IN A 10.10.108.4
c5     IN A 10.10.108.5
c6     IN A 10.10.108.6
c7     IN A 10.10.108.7
c8     IN A 10.10.108.8
c9     IN A 10.10.108.9
c10    IN A 10.10.108.10
c11    IN A 10.10.108.11
c12    IN A 10.10.108.12
www IN CNAME master
ftp IN CNAME master
pop IN CNAME master
```

Утилиты для проверки синтаксических ошибок



В составе сервера BIND поставляются две утилиты, предназначенные для обнаружения синтаксических ошибок в файле named.conf и в файлах описания зон.

named-checkconf предназначена для проверки синтаксических ошибок в файле named.conf. Программе не требуется передавать дополнительные параметры при запуске. Если синтаксические ошибки не будут обнаружены, программа ничего не выведет на экран. Если ошибки будут обнаружены, программа выводит номер строки, в которой найдена ошибка.

Для проверки наличия синтаксических ошибок в файле описания зоны используется программа named-checkzone.

```
named-checkzone домен файл_описания
```

Например, чтобы проверить синтаксис файла описания зоны unix.specialist.ru, необходимо выполнить следующую команду:

```
named-checkzone unix.specialist.ru  /var/named/master.unix.specialist.ru
```

Если в файле будут найдены синтаксические ошибки, будут показаны номера строк, на которых они обнаружены.

Включать или перезапускать DNS сервер рекомендуется только после проверки его конфигурационных файлов на наличие ошибок.

Управление DNS сервером

Программа rndc
Стартовые скрипты системы инициализации
Сигналы

Запускать, останавливать и использовать другие возможности DNS сервера можно при помощи различных программ:

- Поставляемой с BIND 9 программой rndc
- При помощи стартовых скриптов системы инициализации
- При помощи сигналов

Управление DNS сервером при помощи программы rndc

С BIND9 поставляется специальная программа управления rndc.

При вызове программы необходимо указывать одну из перечисленных ниже команд.

Команда	***Описание***
stop	Сохранить внесенные изменения и остановить DNS сервер
halt	То же, что и выше, но без сохранения
trace	Увеличить уровень отладки на один. Если сервер работал в обычном режиме, при включении отладки создается файл named.run, в который будет попадать вся отладочная информация
trace level	Установить заданный уровень отладки
notrace	Выключить режим отладки
flush	Очистить кэш DNS сервера
reload	Перечитать все конфигурационные файлы и файлы описания зон
reload zone [class[view]]	Перечитать определенную зону
reconfig	Перечитать конфигурационный файл и загрузить только новые зоны
stats	Записать статистику сервера в специальный файл named.stats
dumpdb	Сохранить содержимое кэш DNS сервера в файл named_dump.db

Запуск DNS сервера

Для включения DNS сервера необходимо запустить на выполнение демон named. Это можно сделать вручную в командной строке или при помощи стартовых скриптов системы инициализации.

При запуске демона named ему можно указать следующие параметры:

- **-u пользователь** — определяет пользователя, с правами которого он будет запущен. В качестве параметра можно указывать UID или логин пользователя.
- **-t директория** — запустить демон в режиме chroot.

Если использовать стартовые скрипты, то запуск сервера происходит при помощи следующих команд:

Ubuntu Linux:

```
/etc/init.d/bind9 start
```

RedHat Linux:

```
service named start
```

SuSE Linux:

```
/etc/init.d/named start
```

Slackware Linux:

```
/etc/rc.d/rc.bind start
```

Завершение работы DNS сервера

Завершение работы DNS сервера возможно тремя способами.

- При помощи сигнала: `killall named`
- При помощи стартового скрипта, которому в качестве аргумента передается параметр stop.
- При помощи программы rndc: `rndc stop`

Перезагрузка конфигурационных файлов и файлов описания зон

Если демону послать сигнал HUP, он перечитает все конфигурационные файлы и файлы описания зон.

```
killall -HUP named
```

Программа rndc позволяет более гибкое управление DNS сервером.

```
rndc reload
```

Будут перечитаны все конфигурационные файлы и файлы описания зон.

Если необходимо перегрузить только определенную зону, тогда лучше использовать команду reload zone.

```
rndc reload zone class.unix
```

Настройка клиента DNS

Файл /etc/resolv.conf

Параметры:

- nameserver
- search

Для указания клиенту DNS какому серверу он будет посылать запросы, в файле /etc/resolv.conf при помощи параметра nameserver следует указать IP адреса DNS серверов, к которым он должен обращаться.

В этом файле можно определить не более трех параметров nameserver.

Например:

```
nameserver 10.10.108.20
nameserver 10.10.1.1
```

Также в этом файле можно определить один параметр search, при помощи которого можно вводить не полные имена.

В качестве параметров search используют имена доменов, разделенные пробелами.

Например:

```
search u1.unix.specialist.ru unix.specialist.ru
```

Если программе ping в качестве параметра передать имя c1, при передаче его клиенту DNS на преобразование, клиент не обнар жит в нем точек и поймет, что это не FQDN имя. Клиент сначала подставит к имени домен u1.unix.specialist.ru и отправит его на преобразование. Если в ответ он получит сообщение об ошибке, клиент подставит следующий в списке домен.

Лабораторная работа А

Настройка DNS сервера на машине слушателя

Цель работы

Научиться настраивать и управлять DNS сервером.

Исходные данные

В дальнейшем вы будете работать по парам. DNS сервер необходимо настраивать на четных машинах сети. На нечетных машинах будут настроены почтовые сервера.

Вам следует настроить DNS сервер на поддержку домена права на управление которым вам делегировал преподаватель.

Напишите имя домена _____

В файле описания зоны необходимо определить запись MX, указывающую на нечетную машину, на которой будет работать почтовый сервер.

В результате лабораторной работы вы должны преобразовывать имена только ваших машин.

В качестве проверки преобразования воспользуйтесь программой ping.

Задачи	***Описание***
1. Добавление описания зоны в файл named.conf	1. Откройте на редактирование файл /etc/bind/named.conf.local и добавьте в него инструкцию zone, определяющую параметры зоны. 2. Создайте директорию /etc/bind/localnet
2. Создание	В директории /etc/bind/localnet создайте файл описания зоны.
3. Проверка синтаксических ошибок	1. При помощи программы named-checkconf проверьте файл named.conf на наличие синтаксических ошибок. 2. При помощи программы named-checkzone проверьте файл описание зоны на наличие синтаксических ошибок.
4. Запуск DNS сервера	1. Запустите DNS сервер: **`/etc/init.d/bind9 restart`** 2. Посмотрите список процессов и убедитесь, что демон named в нем присутствует. 3. Посмотрите, какую информацию демон named добавил в конец журнального файла /var/log/messages
5. Настройка клиента DNS	На обеих машинах настройте клиента DNS так, чтобы он начал пользоваться вашим DNS сервером.
6. Проверка работоспособности сервера	Выполните программу ping в качестве параметра указывая FQDN имя машины соседа.
7. Изменение имени машины на новое в конфигурационных файлах и перезапуск сервисов	После изменения имени машины, новое имя необходимо записать в конфигурационных файлах и перезапустить сервисы, на которые влияет изменение имени машины.
1. Измените содержимое файлов /etc/hosts и /etc/HOSTNAME	Перегрузите машину и запустите сервер DNS.

Вопросы

1. Для чего в файле named.conf.options необходима инструкция options?

2. При помощи какой инструкции в файле named.conf описываются зоны, поддерживаемые BIND?

3. В каком файле символ «;» является комментарием?

4. Какие типы записей обязательно должны присутствовать в файле описания зоны?

5. Какие утилиты используются для проверки синтаксических ошибок в файле named.conf и файлах описания зоны?

6. Какая программа рекомендуется для управления DNS сервером?

7. Какой файл используется для конфигурации клиента DNS в Linux?

Настройка поддержки slave зоны

```
zone “u1.unix.specialist.ru” IN {
        type slave;
        masters { 10.10.108.2; };
        file “slave.u1.unix.specialist.ru”;
};
```

В BIND slave зона настраивается очень просто, достаточно описать инструкцию zone в файле named.conf.

```
zone «u1.unix.specialist.ru» IN {
       type slave;
       masters { 10.10.108.2; };
       file «slave.u1.unix.specialist.ru»;
};
```

Параметр masters является обязательным. Он определяет IP адрес master сервера, куда наш DNS сервер будет обращаться за информацией о зоне.

Параметр file не является обязательным, но его желательно описывать. В этом параметре определяется файл, в котором будет храниться информация о зоне. Если файл не определен, то информация будет храниться только в оперативной памяти и при перезапуске DNS сервер будет вынужден снова получить содержимое зоны с master сервера.

Лабораторная работа Б

Настройка slave зоны

Цель работы

Научиться поднимать поддержку slave зоны в BIND.

Исходные данные

Slave зона будет подниматься на нечетной машине.

Задачи	***Описание***

1. Настройка политики разрешений AppArmor для DNS-сервера	1. Откройте на редактирование файл
	`/etc/apparmor.d/usr.sbin.named`
	и добавьте право на запись для директории
	`/etc/bind/localnet`
	затем перечитайте настройки AppArmor
	`/etc/init.d/apparmor force-reload`

2. Добавление описания зоны в файл named.conf.local	1. Откройте на редактирование файл `/etc/bind/named.conf.local` и добавьте в него инструкцию zone определяющую параметры зоны.
3. Проверка синтаксиса	1. При помощи программы named-checkconf проверьте файл named.conf на наличие синтаксических ошибок
4. Запуск DNS сервера	1. Запустите DNS сервера `rndc reload` 2. Посмотрите содержимое файла, в котором содержится информация о зоне

Делегирование прав на управление доменом

```
domain.com.
IN NS ns.domain.com.
ns.domain.com.  IN  A   1.2.3.4
```

В этой главе будет рассказано о том, какие действия необходимо предпринять для получения прав на домен.

В первую очередь необходимо обратиться к хозяину домена, в котором вы собираетесь создавать свой поддомен. Например, если вы хотите использовать домен u1.unix.specialist.ru, вы должны обратиться к хозяину домена unix.specialist.ru и узнать условия получения прав на домен u1.unix.specialist.ru.

Вы можете без согласия хозяина вышестоящего домена поднять поддержку домена u1.unix.specialist.ru. Но в этом случае ваш домен будет виден только клиентам, использующим ваш DNS сервер. Всем остальным пользователям Интернет он не будет доступен. Поэтому получение разрешения на создание домена у хозяина вышестоящего в иерархии домена обязательно.

Если хозяин вышестоящего домена согласен делегировать вам права на управление доменом u1.unix.specialist.ru, вы должны выполнить следующие действия:

- На своем DNS сервере поднять поддержку домена
- Договориться с хозяином другого DNS сервера на поддержку slave зоны на их сервере. Желательно иметь несколько slave DNS серверов
- Сообщить хозяину вышестоящего домена IP адреса и имена DNS серверов, ответственных за ваш домен

После этого хозяин вышестоящего домена делегирует вам права на управление доменом. И он становиться доступным всем остальным пользователям Интернет.

Делегирование прав на домен осуществляется при помощи записей типа NS и A, добавляемых в файл описания вышестоящей зоны.

Например, вы предоставили следующую информацию хозяину вышестоящего домена:

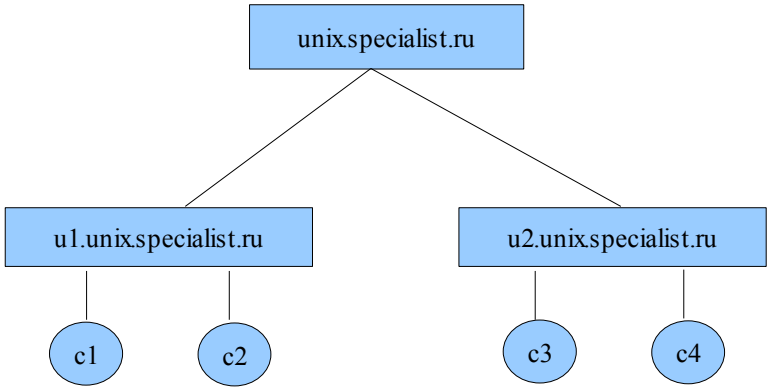
- master DNS сервер — c2.u1.unix.specialist.ru , его IP адрес — 10.10.100.2
- slave DNS сервер — ns.any.com, его IP адрес — 1.2.3.4

В файл описания зоны unix.specialist.ru будут добавлены следующие записи:

```
u1.unix.specialist.ru.    IN NS c2.u1.unix.specialist.ru.
c2.u1.unix.specialist.ru. IN  A 10.10.100.2
u1.unix.specialist.ru.    IN NS ns.any.com.
ns.any.com.               IN  A 1.2.3.4
```

После добавления этих записей DNS сервер, отвечающий за зону unix.specialist.ru, сможет выдать IP DNS серверов, отвечающих за зону u1.unix.specialist.ru.

Использование внутренних доменов



Иногда возникает необходимость создания доменов, используемых для внутренних целей. Этим доменам обычно дают любые имена, не придерживаясь стандартной иерархии. К таким доменам относится и unix.specialist.ru, который мы используем в нашем классе.

Основной недостаток этих доменов — невозможно стандартными средствами получить доступ к ним по цепочке от корневого домена. Если обратиться к DNS серверу, обслуживающему домен specialist.ru, с вопросом о машине master.unix.specialist.ru, в ответ мы получим сообщение об ошибке, поскольку администратор домена specialist.ru не делегировал нам права на управление доменом unix.specialist.ru.

Если в организации стоит всего один DNS сервер, ответственный за домен для внутреннего использования, и все клиенты настроены на работу с этим сервером, никаких проблем не возникнет. Но если, как в нашем классе, будет использоваться несколько доменов и DNS серверов, возникают проблемы в разрешении адресов.

После поднятия DNS серверов на своих машинах, вы можете преобразовывать имена машин только из своего домена. Поскольку при попытке преобразования имени машины из домена unix.specialist.ru или из домена соседней пары, ваш DNS сервер обращается к корневому серверу и в ответ получает сообщение об ошибке.

В приведенной схеме DNS сервер, отвечающий за зону unix.specialist.ru, делегировал права на управление доменами u1.unix.specialist.ru и u2.unix.specialist.ru другим DNS серверам.

Поэтому, если обратиться к нему с вопросом о машине, например, c1.u1.unix.specialist.ru, он знает IP адрес DNS сервера, отвечающего за соответствующую зону и может послать ему запрос о интересующей машине. Поэтому необходимо сделать так, чтобы дополнительные DNS сервера с вопросами о зонах, для которых они не авторитетны (зоны не прописаны в файле named.conf при помощи инструкции zone), обращались к DNS серверу master.unix.specialist.ru. Сделать это можно несколькими способами.

Параметр forwarders

```
forwarders { адрес; адрес; ... };
```

Первый способ наиболее распространен. Когда у вас много DNS серверов, отвечающих за зоны одного уровня, права на которые вам делегированы DNS сервером, отвечающим за вышестоящий в иерархии домен. Например, такие как в нашем классе.

Эти DNS сервера предназначены только для поддержки клиентов одного отдела или подразделения и больше никаких функций на себе не несут.

Для решения проблемы достаточно в конфигурационном файле /etc/named.conf в инструкции options добавить следующую строку:

```
forwarders { 10.10.108.20; };
```

Этот параметр означает, что запросы о всех зонах, для которых текущий DNS сервер не авторитетен, будут пересылаться на сервер с указанным IP адресом. В этом случае наш DNS сервер самостоятельно не будет посылать запросы на корневой сервер и система начнет работать нормально.

Зона типа forward

```
zone "u1.class.unix" IN {
type forward;
forwarders { 10.10.100.2; };
};
```

Предположим, что у нас на выходе в интернет существует DNS сервер, задача которого кэшировать все запросы от DNS серверов внутренней сети предприятия. И на firewall стоит разрешение на посылку запросов к DNS серверам, расположенным в Интернет только с этого кэширующего сервера.

Чтобы DNS сервер, отвечающий за зону unix.specialist.ru, мог нормально обслуживать запросы на преобразование имен машин, расположенных в интернет, ему необходимо иметь возможность посылать запросы к DNS серверам корневого домена и других доменов интернет. Но на firewall прямой доступ к ним закрыт.

Для того, чтобы все работало, в файле named.conf этого DNS сервера тоже необходимо прописать строку forwarders, со ссылкой на кэширующий DNS сервер предприятия:

```
forwarders { 10.10.1.1; };
```

И, казалось бы, система должна заработать нормально. Но, к сожалению этого не произойдет. Поскольку мы прописали параметр forwarders, DNS сервер, отвечающий за зону unix.specialist.ru, станет отвечать только на вопросы о зонах для которых он авторитетен, то есть, описанных при помощи инструкции zone. Поскольку домены u1.unix.specialist.ru , u2.unix.specialist.ru и т.д. не описаны при помощи инструкции zone, этот DNS сервер будет пересылать запросы о них на DNS сервер, определенный при помощи параметра forwarders, даже не смотря на то, что он сам делегировал права на управление ими. А кэширующий DNS сервер, просто переправит этот запрос на корневой DNS сервер и получит сообщение об ошибке.

Для решения этой проблемы в файле named.conf DNS сервера, отвечающего за зону unix.specialist.ru следует описать зоны u1.unix.specialist.ru, u2.unix.specialist.ru и другие, при помощи инструкции zone. Использовать тип master нельзя, потому что наш DNS сервер не является master сервером для этих зон. Возможно поднять slave зоны, но это не всегда является правильным решением. В нашей ситуации наиболее правильным выбором будет описание зоны типа forward. Например, для зоны u1.unix.specialist.ru следует добавить такие строки:

```
zone «u1.unix.specialist.ru» IN {
        type forward;
        forwarders { 10.10.108.1; };
};
```

Теперь, если на наш DNS сервер придет запрос на преобразование имен компьютеров из домена u1.unix.specialist.ru, он перешлет этот запрос на DNS сервер, описанный при помощи параметра forwarders в инструкции zone.

Вопросы

1. При помощи какого параметра можно заставить DNS сервер пересылать все вопросы о зонах, для которых он не авторитетен на DNS сервер провайдера?

2. Для чего используется зона типа forward?

Тестирование и отладка DNS сервера

Просмотр содержимого журнальных файлов
Отладочный режим сервера. Программы посылки запросов

Для тестирования и отладки DNS сервера можно использовать несколько способов:

- Просмотр содержания журнальных файлов
- Запуск сервера в отладочном режиме
- Использование программ, посылающих запросы DNS серверу

При старте демон named помещает в журнальные файлы различную информацию:

- Интерфейсы, на которых демон слушает запросы и команды
- Какие зоны были загружены или перезагружены

К сожалению, информация, которая поступает в журнальные файлы, не является подробной и иногда ее не хватает для определения ошибок работы DNS сервера.

BIND позволяет выбирать, какая информация и в каком объеме должна попадать в журнальные файлы. Для этого используется инструкция logging. Но ее применение не оправдано из-за больших затрат времени на настройку. Кроме того, отладочный режим сервера необходим достаточно редко. Поэтому для отладки следует использовать программу rndc и/или программы посылающие, запросы серверу DNS из командной строки.

Использование уровней отладки DNS сервера

Уровни отладки:

- 0 — выключить отладку
- 1-2 — отладка конфигурационных файлов. Остальные уровни используются разработчиками BIND

```
rndc trace rndc notrace
```

Уровни отладки DNS сервера обозначаются числами от 0 до

Чем больше число, тем более подробную информацию будет содержать вывод программы.

Уровень 0 выключает отладочный режим. Уровни 1 и 2 используются для отладки конфигурационных файлов и файлов описания зон. Остальные уровни отладки используются разработчиками BIND.

Самый простой способ включения отладочного режима — запуск демона named с опцией -d.

```
named -d2
```

После запуска демона вся отладочная информация будет помещаться в файл / var/named/named.run.

Включение отладки можно производить при помощи утилиты rndc.

```
rndc trace 2
```

В этом случае тоже появляется файл named.run.

Для отключения отладочного режима тоже следует использовать rndc:

```
rndc notrace
```

В любом случае, прежде чем использовать режимы отладки, настоятельно

рекомендуется произвести поиск синтаксических ошибок при помощи программ named-checkconf и named-checkzone.

Использование программ nslookup, dig и host

Программы:

- nslookup
- dig
- host

Для посылки запросов DNS серверу можно использовать следующие программы:

- nslookup
- dig — программа имеет ту же функциональность, что и nslookup, но параметры по умолчанию этой программы более эффективны. Она выдает больше информации и является более удобной в использовании
- host — программа выдает краткую информацию

Программа nslookup

Программа может работать в двух режимах:

- Все параметры задаются в командной строке
- Интерактивный режим

Ниже приведены некоторые команды, которые можно использовать в интерактивном режиме.

Инструкция	***Описание***
exit	Выход из программы
server	Выбор сервера по умолчанию, к которому будут посылаться запросы. По умолчанию используется DNS сервер, описанный в файле /etc/resolv.conf
set type=тип	Установка типа записи для запроса

Программа dig

Программа dig не имеет интерактивного режима работы, все ответы она выдает на стандартный вывод в формате файла описания зоны.

```
dig [опции] имя [тип записи] [@DNS_сервер]
```

Программе dig необходимо указать имя машины или домена, информацию о которых вы хотите получить. Так же можно определить тип интересующей вас записи.

Программа по умолчанию посылает запрос DNS серверу, определенному в файле /etc/resolv.conf. Если необходимо послать запрос какому-либо другому серверу, его IP адрес указывается после символа @.

Если вы хотите получить информацию об обратном преобразовании из IP в имя машины, при вызове программы необходимо использовать опцию -x.

Программа host

Программа выдает краткую информацию. При ее вызове необходимо указать имя интересующей машины.

На что следует обращать внимание при работе с перечисленными выше программами? В первую очередь на то, какие ответы выдает DNS сервер. Обычно таким образом определяются имена машин и доменов у которых в конце имени не стоит корневой домен.

Если программа не может получить ответ от DNS сервера, значит вы неправильно определили его в файле /etc/named.conf или по каким то причинам сервер недоступен.

Вопросы

1. Какие уровни отладки рекомендуется использовать администратору для поиска ошибок в работе DNS сервера.

2. В какой файл будет попадать отладочная информация при включении режима отладки?

3. Как выключить режим отладки?

Зоны обратного преобразования

Домен in-addr.arpa

Все примеры, которые мы до сих пор рассматривали, относились к прямому преобразованию, когда имя машины преобразовывалось в IP адрес. DNS сервера позволяют осуществлять и обратное преобразование — IP адрес в имя машины.

Для того, чтобы понять, как эта возможность реализована, необходимо понять принцип образования поддоменов в домене .in-addr.arpa. Это специальный домен, предназначенный для обратного преобразования.

Если посмотреть на имя машины master.unix.specialist.ru, то видно, что оно состоит из двух частей: master — имя, unix.specialist.ru

— домен.

IP адрес этой машины (предположим, что его значение равно 10.10.108.20, netmask 255.255.255.0) тоже состоит из двух частей:

10.10.108 — адрес сети, 20 — адрес машины в сети. Если этот IP адрес записать наоборот: 20.800.10.10 — то можно представить что 20 — это имя машины, 108.10.10 — имя домена.

С учетом использования специального домена in-addr.arpa, FQDN будет выглядеть следующим образом:

```
20.108.10.10.in-addr.arpa
```

Таким образом, для преобразования IP адресов в сети 10.10.108/24 в файле named.conf необходимо определить зону 108.10.10.in-addr.arpa. Она может быть как master так и slave. В файле описания зоны обязательно должны быть определены записи SOA и NS. Для преобразования IP адресов используется запись типа PTR.

```
имя_ip IN PTR каноническое_имя
```

Для преобразования IP адреса 10.10.100.20 в файле описания зоны необходимо добавить следующую запись:

```
20 IN PTR master.unix.specialist.ru.
```

Зоны обратного преобразования должен поднимать владелец пула IP адресов. Если провайдер выделил вам не весь пул цел ком, вы должны передать ему имена машин для того, чтобы он внес их в зону обратного преобразования.

Лабораторная работа В

Настройка зоны обратного преобразования

Цель работы

Научиться поднимать поддержку зоны обратного преобразования в BIND.

Исходные данные

Зона обратного преобразования будет подниматься на всех четных машинах (машинах, на которых расположены DNS сервера ваших зон).

В зоне обратного преобразования достаточно описать только три машины: две машины, входящие в ваш домен и машину пр подавателя.

Задачи	***Описание***
1. Добавление описания зоны в файл named.conf	1. Откройте на редактирование файл /etc/bind/named.conf.local и добавьте в него инструкцию zone определяющую параметры зоны.
2. Создание файла описания зоны	1. Создайте файл описания зоны обратного преобразования.
3. Проверка синтаксиса	1. При помощи программы named-checkconf проверьте файл named.conf на наличие синтаксических ошибок 2. При помощи программы named-checkzone проверьте файл зоны обратного преобразования на наличие ошибок
4. Включение поддержки зоны обратного преобразования	1. Заставьте ваш DNS сервер перечитать свои конфигурационные файлы. **rndc reload** 2. Посмотрите, какая информация была добавлена в конец журнального файла /var/log/messages
5. Проверка работоспособности	1. Выполните программу nslookup, указав ей в качестве параметра IP адрес машины преподавателя 2. Выполните программу dig, указав ей в качестве параметров опцию -x и IP адрес машины преподавателя

Если у вас возникли ошибки во время выполнения лабораторной работы, ниже опишите возникшие ошибки и способы их решения.

Вопросы

1. Какой тип записи используется для преобразования IP адреса в имя машины?

2. Кто должен поднимать зоны обратного преобразования?

Кэширующий сервер

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

Довольно часто DNS сервера используются как простые кэширующие сервера без поддержки каких-либо зон. Такие сервера позволяют экономить достаточно большое количество трафика.

Для того, чтобы BIND стал кэширующим DNS сервером, в конфигурационном файле сервера должны быть описаны три зоны:

- Зона прямого преобразования для домена localhost
- Зона обратного преобразования 0.0.127.in-addr.arpa
- Зона, описывающая корневой домен

Для нормальной работы кэширующего DNS сервера ему необходимо каким-либо образом указать IP адреса DNS серверов, отвечающих за корневой домен. Потому что он, если информации о машине нет в локальном кэше, будет сначала обращаться к корневому домену и по иерархии DNS серверов добираться до сервера, ответственного за интересующую зону.

IP адреса корневых DNS серверов описывают при помощи специальной зоны типа hint. Ее основное назначение “подсказать” нашему серверу эти IP адреса, чтобы он мог обратиться к корневым серверам.

```
zone "." IN {
        type hint;
        file "caching-examples/named.ca";
};
```

Обязательным параметром при определении зоны типа hint является параметр file, указывающий на файл, в котором хранятся подсказки.

В этом файле содержатся пары записей типа NS и A, определяющие DNS сервера, отвечающие за корневой домен:

```
.3600000 IN  NS A.ROOT-SERVERS.NET. A.ROOT-SERVERS.NET.    3600000       A      198.41.0.4
```

Если вы случайно удалили файл подсказок, его можно получить с другого DNS сервера при помощи утилиты dig:

```
dig . ns @IP_DNS > file
```

или с ftp сервера ftp.internic.net/domain/named.cache.

Вопросы

1. Какой тип зоны используется для указания нашему DNS серверу адресов серверов, отвечающих за корневую зону?

2. Как можно получить файл, используемый в качестве подсказки о именах и IP адресах DNS серверов, отвечающих за корневую зону?

Вопросы безопасности

Ограничение доступа к DNS серверу

Технология TSIG

Запуск DNS сервера в chroot

При использовании DNS сервера BIND вы должны представлять, какие потенциальные опасности могут возникнуть в процессе его эксплуатации.

Эти опасности следует разделить на две части:

- Проблемы системы DNS
- Проблемы программы named

Ограничение доступа к DNS серверу

allow-transfer — ограничение зонных пересылок

allow-update — ограничение изменений в описании зоны

allow-query — ограничение запросов к DNS серверу

Первая разновидность проблем связана с изначальной открытостью системы DNS. У злоумышленника есть несколько возможностей использования DNS для организации атак:

- Получение информации о сети предприятия
- Замена информации на DNS серверах

Ограничение доступа к информации

Злоумышленнику для того, чтобы понять, как атаковать вашу сеть, необходимо узнать топологию сети и определить, какие машины какие функции выполняют. Эти параметры можно вычислить на основании записей типа A.

Защита от первого типа уязвимости уже встроена в BIND — нельзя получить все записи типа A с DNS сервера при помощи стандартных программ типа nslookup и dig.

Но если вы заметили, когда была поднята slave зона на нечетных машинах, DNS сервер получил полную информацию о зоне, несмотря на то, что его не было в списке авторитетных серверов, определенных при помощи записи NS.

Для решения этой проблемы в файле named.conf следует добавить параметр allow-transfer, ограничивающий возможность зонных пересылок. В качестве аргументов ему необходимо указывать список IP адресов salve серверов.

```
allow-transfer { 1.2.3.4; 10.10.10.10; };
```

Параметр можно определять в инструкции options. Тогда он будет общим для всех зон вашего DNS сервера. Или определять его для каждой зоны отдельно.

Если предполагается, что DNS сервер будет использоваться только для внутренних нужд, или только некоторые зоны предназначены для доступа к ним из Интернет, BIND позволяет ввести ограничения на то, с каких сетей или IP адресов можно обращаться с запросами к нашему DNS серверу.

Для такого ограничения используют параметр allow-query.

```
allow-query { 10.10.100/24; 1.2.3.4; };
```

Его можно определять в инструкции options, тогда это будут ограничения на запросы к DNS серверу вообще. Если его определить внутри зоны, тогда ограничения будут

распространяться только на эту зону.

Подмена информации на DNS серверах

Эта проблема может возникнуть из-за поддержки расширенного протокола DNS, в котором предусмотрена возможность автоматического обновления информации в зоне DHCP серверами.

По умолчанию изменение информации в зоне возможна со всех машин. Для определения списка машин, откуда можно изменять содержимое зоны, используется параметр allow-update.

```
allow-update { 1.2.3.4; };
```

Если не предполагается использовать возможность изменения информации в зоне, эту возможность необходимо запрещать при помощи ключевого слова none.

```
allow-update { none; };
```

Использование технологии TSIG

```
Программа dnssec-keygen
key “c1-c2” {
     algorithm hmack-md5;
     secret “y3AukehA+ji36hTFTDx3MA==”;
};
server c2.u1.class.unix {
     keys { c1-c2; };
};
allow-transfer { key c1-c2; };
```

Когда мы говорим о параметрах, описанных в предыдущей главе, необходимо понимать, что контроль по IP адресам не самый надежный метод защиты. Если в сети существует DHCP сервер, которому разрешено изменять информацию в зоне, злоумышленник может воспользоваться этим, используя его IP адрес. Так же существует вероятность изменения информации при пересылке зоны на slave сервера.

Для решения этих проблем можно использовать технологию TSIG, которая позволяет при пересылках информации аутентифицировать пару отправитель-получатель и проверять, изменялись ли данные при передаче.

В TSIG применяется метод симметричного шифрования. Для каждой пары серверов, между которыми предполагается передавать данные, должен создаваться свой ключ.

TSIG можно использовать только при передаче данных между серверами. Клиенты DNS не могут использовать эту технологию.

Сигнатуры транзакций, используемые в TSIG, проверяются на момент поступления данных и тут же отбрасываются, то есть, они не хранятся в кэш сервера.

Для генерации ключей используется программа dnssec-keygen, поставляемая с BIND 9. Предположим, что у нас используются два сервера: c1.u1.unix.specialist.ru и c2.u1.unix.specialist.ru, тогда для генерации ключа необходимо выполнить следующую команду:

```
dnssec-keygen -a HMAC-MD5 -b 128 -n HOST c1.u1.unix.specialist.ru  \
-c2.u1.unix.specialist.ru
```

В результате будут созданы два файла: Kc1.u1.unix.specialist.ru - c2.u1.unix.specialist.ru.+157+29237.private и Kc1.u1.unix.specialist.ru - c2.u1.unix.specialist.ru.+157+29237.key, где 157 — код алгоритма MD5, 29237 — случайное число, используемое в качестве идентификатора ключа на случай, когда у

одной пары серверов есть несколько ключей.

Содержимое файла Kc1.u1.unix.specialist.ru -c2.u1.unix.specialist.ru.+157+29237.private может выглядеть так:

```
Private-key-format: v1.2
Algorithm: 157 (HMAC_MD5)
Key: y3AukehA+ji36hTFTDx3MA==
```

Файл Kc1.u1.unix.specialist.ru- c2.u1.unix.specialist.ru.+157+29237.key использоваться не будет и г нерируется программой dnssec-keygen только потому, что она так же применяется для генерирования пар открытых и закрытых ключей.

Ключ должен быть одинаковым на обеих серверах. Установите на него права доступа 600 и передайте пользователю, с правами которого запускается демон named.

Ни в коем случае не посылайте файл по электронной почте или ftp в открытом виде, это не безопасно. Переносите его на дискете или перед передачей зашифруйте программой gpg.

Для определения ключа в BIND в директории /etc/bind создайте файл, например, c1-c2.key. Содержимое этого файла будет следующим:

```
key “c1-c2” {
algorithm hmack-md5;
secret “y3AukehA+ji36hTFTDx3MA==”;
};
```

Доступ к этому файлу должен иметь только пользователь, с правами которого запускается демон named. Включить этот файл в основной конфигурационный файл named.conf можно при помощи инструкции include:

```
include “c1-c2.key”;
```

После определения ключа необходимо заставить named идентифицировать другой сервер:

```
server c2.u1.unix.specialist.ru {
keys { c1-c2; };
};
```

Точно так же необходимо описать аутентификацию нашего DNS сервера на сервере c2.u1.unix.specialist.ru.

Все директивы allow-update, allow-query и allow-transfer должны ссылаться на этот ключ.

Например:

```
allow-transfer { key c1-c2; };
```

Запуск DNS сервера с правами обыкновенного пользователя

Параметр -u

Если запускать DNS сервер из командной строки пользователем root — сервер будет работать с правами суперпользователя.

Это не правильно. Для работы серверу достаточно иметь права обыкновенного пользователя.

Запустить сервер с правами другого пользователя очень просто, достаточно указать параметр -u и логин или UID пользователя.

Если посмотреть стартовый скрипт Ubuntu Server

```
/etc/init.d/bind9
```

видно, что сервер запускается с правами суперпользователя. Что бы это исправить

достаточно добавить параметр -u в переменную OPTIONS используемой для запуска демона:

```
OPTIONS= «-u bind»
```

Но после изменения пользователя не забудьте передать ему директории /var/named и /var/run/named со всем содержимым.

Запуск DNS сервера в chroot

```
named -t /var/named/chroot
```

К сожалению, довольно часто наблюдались случаи взлома серверов через BIND. Для уменьшения возможностей взлома, всегда необходимо обновлять версии BIND. Ни в коем случае не запускать его с правами пользователя root. И, по возможности, запускать его при помощи программы chroot.

Программа chroot заменяет программе корневую директорию на другую. После этого запускаемая при помощи chroot программа не «видит» файловую систему за пределами указанной ей директории.

Но поскольку она не видит другие директории, приходится переносить все необходимые для работы этой программы файлы в ту директорию, которая будет выступать в роли корневой. Причем в этой директории необходимо создать такое же дерево необходимых для работы директорий, как и в корне.

При запуске named можно использовать опцию -t с указанием директории, которая будет выступать в роли корневой директории. То есть, запускать DNS сервер можно и без явного вызова программы chroot.

Если вы хотите запускать named в chroot, следует выполнить следующие действия:

1. Запускать сервер с параметром -t /var/named

2. В директории /var/named создать все необходимые файлы и директории (какие файлы необходимо создать подробно описано в Chroot-BIND-HOWTO)

3. Перенести файл named.conf в директорию /var/named/etc и создать на него символьную ссылку /etc/named.conf

4. Перенести все файлы описания зон в директорию /var/named/var/named

Вопросы

1. Какой параметр необходимо обязательно использовать в инструкции zone для ограничения возможности зонных пересылок?

2. Какой параметр необходимо использовать в инструкции zone для запрещения изменения описания зоны?

3. Можно ли использовать технологию TSIG при взаимодействии клиента и сервера DNS?

4. По каким причинам необходимо запускать DNS сервер в chroot?

DHCP

DHCP (англ. Dynamic Host Configuration Protocol — протокол динамической конфигурации узла) — это сетевой протокол, позволяющий компьютерам автоматически получать IP-адрес и другие параметры, необходимые для работы в сети TCP/IP. Для этого компьютер обращается к специальному серверу, называемому сервером DHCP. Сетевой администратор может задать диапазон адресов, распределяемых среди компьютеров. Это позволяет избежать ручной настройки компьютеров сети и уменьшает количество ошибок. Протокол DHCP используется в большинстве крупных сетей TCP/IP.

DHCP является расширением протокола BOOTP, использовавшегося ранее для обеспечения бездисковых рабочих станций IP-адресами при их загрузке. DHCP сохраняет обратную совместимость с BOOTP.

История

Стандарт протокола DHCP был принят в октябре 1993 года. Действующая версия протокола (март 1997 года) описана в RFC 2131. Новая версия DHCP, предназначенная для использования в среде IPv6, носит название DHCPv6 и определена в RFC 3315 (июль 2003 года).

Распределение IP-адресов

Протокол DHCP предоставляет три способа распределения IP-адресов:

- **Ручное распределение.** При этом способе сетевой администратор сопоставляет аппаратному адресу (обычно MAC-адресу) каждого клиентского компьютера определённый IP-адрес. Фактически, данный способ распределения адресов отличается от ручной настройки каждого компьютера лишь тем, что сведения об адресах хранятся централизованно (на сервере DHCP), и потому их проще изменять при необходимости.
- **Автоматическое распределение.** При данном способе каждому компьютеру на постоянное использование выделяется произвольный свободный IP-адрес из определённого администратором диапазона.
- **Динамическое распределение.** Этот способ аналогичен автоматическому распределению, за исключением того, что адрес выдаётся компьютеру не на постоянное пользование, а на определённый срок. Это называется арендой адреса. По истечении срока аренды IP-адрес вновь считается свободным, и клиент обязан запросить новый (он, впрочем, может оказаться тем же самым).

Некоторые реализации службы DHCP способны автоматически обновлять записи DNS, соответствующие клиентским компьютерам, при выделении им новых адресов. Это производится при помощи протокола обновления DNS, описанного в RFC 2136.

Опции DHCP

Помимо IP-адреса, DHCP также может сообщать клиенту дополнительные параметры, необходимые для нормальной работы в сети. Эти параметры называются опциями DHCP. Список стандартных опций можно найти в RFC 2132.

Некоторыми из наиболее часто используемых опций являются:

- IP-адрес маршрутизатора по умолчанию

- маска подсети
- адреса серверов DNS
- имя домена DNS

Некоторые поставщики программного обеспечения могут определять собственные, дополнительные опции DHCP.

Логика работы протокола DHCP

Протокол DHCP является клиент-серверным, то есть в его работе участвуют клиент DHCP и сервер DHCP. Передача данных производится при помощи протокола UDP, при этом сервер принимает сообщения от клиентов на порт 67 и отправляет сообщения клиентам на порт 68.
Пример процесса получения адреса

Рассмотрим пример процесса получения IP-адреса клиентом от сервера DHCP. Предположим, клиент ещё не имеет собственного IP-адреса, но ему известен его предыдущий адрес — 192.168.1.100. Процесс состоит из четырёх этапов.

Обнаружение DHCP

В начале клиент выполняет широковещательный запрос по всей физической сети с целью обнаружить доступные DHCP-серверы. Он отправляет сообщение типа DHCPDISCOVER, при этом в качестве IP-адреса источника указывается 0.0.0.0 (так как компьютер ещё не имеет собственного IP-адреса), а в качестве адреса назначения — широковещательный адрес 255.255.255.255.

Клиент заполняет несколько полей сообщения начальными значениями:

- уникальный идентификатор транзакции, который позволяет отличать данный процесс получения IP-адреса от других, протекающих в то же время
- аппаратный адрес (MAC-адрес) клиента
- последний известный клиенту IP-адрес, а в данном примере это 192.168.1.100. Это необязательно и может быть проигнорировано сервером.

Сообщение DHCPDISCOVER может быть распространено за пределы локальной физической сети при помощи специально настроенных агентов ретрансляции DHCP, перенаправляющих поступающие от клиентов сообщения DHCP серверам в других подсетях.

Предложение DHCP

Получив сообщение от клиента, сервер определяет требуемую конфигурацию клиента в соответствии с указанными сетевым администратором настройками. В данном случае DHCP-сервер согласен с запрошенным клиентом адресом 192.168.1.100. Сервер отправляет ему ответ (DHCPOFFER), в котором предлагает конфигурацию.

Это сообщение DHCP-сервер рассылает широковещательно. Клиент может получить несколько различных предложений DHCP от разных серверов; из них он должен выбрать то, которое его «устраивает».

Запрос DHCP

Выбрав одну из конфигураций, предложенных DHCP-серверами, клиент отправляет запрос DHCP (DHCPREQUEST). Он рассылается широковещательно; при этом к опциям, указанным клиентом в сообщении DHCPDISCOVER, добавляется специальная опция — идентификатор сервера — указывающая адрес DHCP-сервера, выбранного клиентом (в данном случае — 192.168.1.1).

Подтверждение DHCP

Наконец, сервер подтверждает запрос и направляет это подтверждение (DHCPACK) клиенту. После этого клиент должен настроить свой сетевой интерфейс, используя предоставленные опции.

Прочие сообщения DHCP

Помимо сообщений, необходимых для первоначального получения IP-адреса клиентом, DHCP предусматривает несколько дополнительных сообщений для выполнения иных задач.

Отказ DHCP

Если после получения подтверждения (DHCPACK) от сервера клиент обнаруживает, что указанный сервером адрес уже используется в сети, он рассылает широковещательное сообщение отказа DHCP (DHCPDECLINE), после чего процедура получения IP-адреса повторяется. Использование IP-адреса другим клиентом можно обнаружить, выполнив запрос ARP.

Отмена DHCP

Если по каким-то причинам сервер не может предоставить клиенту запрошенный IP-адрес, или если аренда адреса удаляется администратором, сервер рассылает широковещательное сообщение отмены DHCP (DHCPNAK). При получении такого сообщения соответствующий клиент должен повторить процедуру получения адреса.

Освобождение DHCP

Клиент может явным образом прекратить аренду IP-адреса. Для этого он отправляет сообщение освобождения DHCP (DHCPRELEASE) тому серверу, который предоставил ему адрес в аренду. В отличие от других сообщений DHCP, DHCPRELEASE не рассылается широковещательно.

Информация DHCP

Сообщение информации DHCP (DHCPINFORM) предназначено для определения дополнительных параметров TCP/IP (например, адреса маршрутизатора по умолчанию, DNS-серверов и т. п.) теми клиентами, которым не нужен динамический IP-адрес (то есть адрес которых настроен вручную). Серверы отвечают на такой запрос сообщением подтверждения (DHCPACK) без выделения IP-адреса.

Настройка DHCP

Файлы конфигурации dhcp-сервера:

- /etc/default/dhcp
- /etc/dhcpd.conf

Начинать настройку dhcp-сервера с файла /etc/default/dhcp, впишите туда имя интерфейса на котором будет работать ваш DHCP сервер, так же надо учесть что в настройках пула раздаваемых клиентам адресов должна быть та же подсеть что и на интерфейсе указанном в файле, иначе сервер не стартует.

Теперь можно настраивать сам сервер, откройте файл /etc/dhcpd.conf и впишите туда нужные вам параметры по аналогии с примером приведенным ниже. В файле конфигурации установленном по умолчанию приведены некоторые примеры которые могут вам понадобится в будущем, поэтому оставим его на всякий случай и создадим новый файл конфигурации:

```
cp /etc/dhcpd.conf /etc/dhcpd.conf.default
echo -n  > /etc/dhcp3/dhcpd.conf
nano /etc/dhcpd.conf
```

Пример файла /etc/dhcpd.conf:

```
# Определение глобальных настроек
option domain-name "example.ru";
option domain-name-servers 192.168.2.1;
default-lease-time 604800;
max-lease-time 864001;
#
# Здесь указана подсеть адресов для выдачи клиентам, DNS сервера, NETBIOS сервера
# доменное имя, широковещательный адрес, и диапазон выдаваемых адресов.
subnet 192.168.0.0 netmask 255.255.255.0 {
option netbios-name-servers 192.168.2.1;
option domain-name-servers 192.168.2.1;
option domain-name "example.ru";
option broadcast-address 192.168.2.255;
option routers 192.168.2.1;
range 192.168.2.2 192.168.2.250;
}
```

Лабораторная работа Г.

Цель работы

Научиться настраивать DHCP-сервер.

Задачи	*Описание*
1. Подготовка	1. Установите DHCP-сервер **`apt-get install dhcp3-server`** 2. Откройте на редактирование файл **`/etc/default/dhcp`** 3. В переменной, используемой при запуске DHCP-сервера присвойте интерфейс, через который он подключен к соседу **`INTERFACES=«eth1»`** 4. Сохраните файл
2. Настройка сервера	Настройте файл /etc/dhcp3/dhcpd.conf в соответствии с описанным выше примером.
3. Запуск сервера	1. Перезапустите DHCP-сервер **`/etc/init.d/dhcp restart`** 2. Проверьте, запустился ли dhcp-сервер: **`pgrep dhcpd`**
4. Проверка работы сервера	1. На компьютере соседа наберите **`dhclient eth1`** **, где eth1 — интерфейс к соседу** 2. Проверьте, правильно ли происходит присвоение сетевых настроек.