Министерство образования Республики Беларусь
Учреждение образования «Полоцкий государственный университет»

Факультет информационных технологий
Кафедра технологий программирования

Методические указания
к выполнению лабораторной работы №5

по дисциплине «**Компьютерные системы и сети**»
для специальности 1-40 01 01 Программное обеспечение информационных технологий

на тему «**Настройка службы DNS для Ubuntu Server**»

Новополоцк, 2018 г.

Название: «Настройка службы DNS для Ubuntu Server».

Цель работы: Ознакомиться с особенностями настройки службы DNS на Ubuntu Server. Изучить настройку пакета BIND9 для операционной системы Ubuntu 11.04.

Теоретическая часть

Служба DNS

Согласно концепции TCP/IP, каждый хост в сети должен иметь как минимум один IP-адрес. Этот адрес используется для однозначной идентификации этого хоста в сети. Однако, существует недостаток данного метода, заключающийся в субъективности человеческого восприятия. В сети с большим количеством хостов, невозможно запомнить все нужные IP-адреса. Для решения этой проблемы была предложена схема именования, получившая название **системы доменных имен** (DNS – Domain Name System). Имена, используемые DNS для именования хостов, получили название **полных имен домена** (FQDN – Full Qualified Domain Name).

В основе системы доменных имен лежит иерархическое пространство имен.

При этом всё пространство имен DNS представлено в виде отдельных фрагментов, называемых **доменами** (domains). Домены, связываясь между собой при помощи отношений родитель – потомок, образуют определенную иерархию. В зависимости от того, какое положение занимает домен в этой иерархии, принято говорить об уровне домена.

Домен, лежащий в основании иерархического пространства имен DNS, получил название **корневого домена** (root domain). Корневой домен выполняет функцию родоначальника всех доменов первого уровня. Фактически он является чисто формальным элементом, символизирующим иерархичность пространства доменных имен. Для записи доменного имени корневой домен обозначается как пустое место точки, которой заканчивается любое доменное имя.

Домены первого уровня используются для группировки других доменов по организационному признаку либо географическом положению. В случае группировки по организационному уровню имена доменов первого уровня образуются тремя символами(.edu – образовательные учреждения, .com – коммерческие организации, .org – некоммерческие организации). Для определения принадлежности к стране используют имена из двух символов(.ru, .by).

Кроме этого существует еще один домен первого уровня, который используется для группировки **обратных адресов** (reverse domains). Обратные домены применяются для осуществления поиска доменного имени хоста по его IP-адресу. Этот специальный домен получил название **.arpa**, и он является единственным доменом первого уровня, имеющим имя из четырех символов. Домен содержит только один домен второго уровня: **.in-addr.arpa**. Соответственно дочерние домены данного домена имеют вид **168.192.in-addr.arpa** (Для подсети 192.168.0.0/16). Требование наличия

данной зоны обусловлено многими факторами, в частности некоторые программы аутентифицируют входящий траффик по имени адресата.

Домены первого уровня используются исключительно для группировки доменов следующих уровней по некоторому признаку. Пример иерархии DNS в виде дерева предоставлен на рисунке 1.

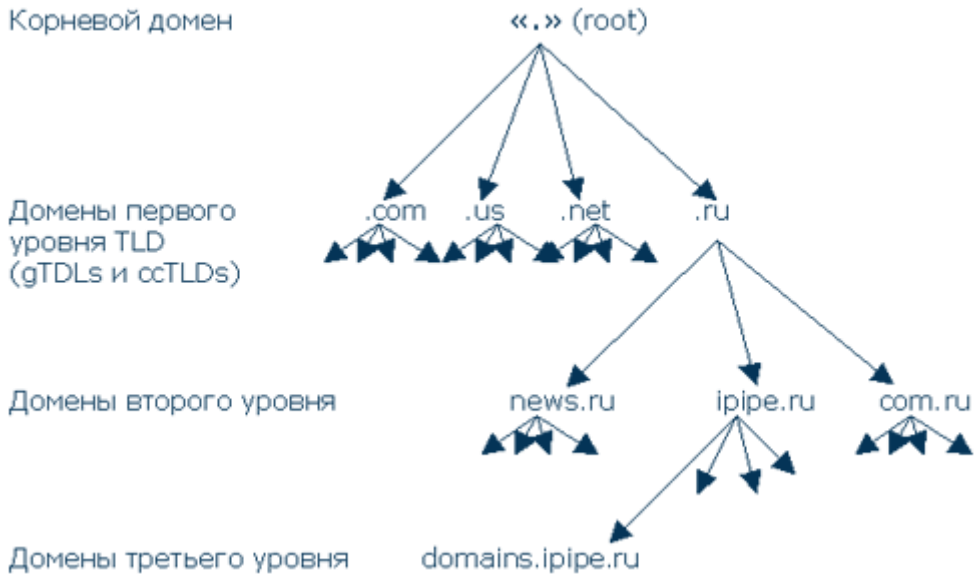

Рисунок 1 – Иерархия DNS в древовидном виде

Каждый из узлов соответствует либо домену, либо хосту. Под хостом в этом дереве понимают лист, т.е. такой узел ниже которого нет других узлов. Именовать хост можно либо частичным именем, либо полным именем. **Полное имя хоста** - это имя, в котором перечисляются слева направо имена всех промежуточных узлов между листом и корнем дерева доменного именования, при этом начинают с имени листа, а кончают корнем, например, *polyn.net.kiae.su.*

Причем, часть полного доменного имени, перечисляющая слева направо имена всех промежуточных узлов между листом и корнем дерева доменного именования, называется **DNS-суффиксом**.

Частичное имя – это имя, в котором перечислены не все, а только часть имен узлов, например:

polyn
apollo.polyn
quest.polyn.kiae

Обратите внимание на то, что в частичных (неполных именах) символ точки в конце имени не ставится. В реальной жизни программное обеспечение системы доменных имен расширяет неполные имена до полных прежде, чем обратиться к серверам доменных мен за IP-адресом.

Слово «Хост» не является в полном смысле синонимом имени компьютера, как это часто упрощенно представляется. Во-первых, у компьютера может быть множество IP-адресов, каждому из которых можно поставить в соответствие одно или несколько доменных имен. Во-вторых, одному доменному имени можно поставить в соответствие несколько разных

IP-адресов, которые, в свою очередь могут быть закреплены за разными компьютерами.

Еще раз обратим внимание на то, что именование идет слева направо, от минимального имени хоста (от листа) к имени корневого домена. Разберем, например, полное доменное имя demin.polyn.kiae.su. Имя хоста – demin, имя домена, в который данный хост входит, – polyn, имя домена, который охватывает домен polyn, т.е. является более широким по отношению к polyn, – kiae, в свою очередь последний (kiae) входит в состав домена su.

Имя polyn.kiae.su – это уже имя домена. Под ним понимают имя множества хостов, у которых в их имени присутствует polyn.kiae.su. Вообще говоря, за именем polyn.kiae.su может быть закреплен и конкретный IP-адрес. В этом случае кроме имени домена данное имя будет обозначать и имя хоста. Такой прием довольно часто используется для обеспечения коротких и выразительных адресов системе электронной почты.

Следует также упомянуть о канонических доменных именах. Это понятие встречается в контексте описания конфигураций поддоменов и зон ответственности отдельных серверов доменных имен. С точки зрения дерева доменных имена не разделяют на канонические и неканонические, но с точки зрения администраторов, серверов и систем электронной почты такое разделение является существенным. Каноническое имя - это имя, которому в соответствие явно поставлен IP-адрес, и которое само явно поставлено в соответствие IP-адресу. Неканоническое имя - это синоним канонического имени.

Как работает DNS?

DNS работает в режиме вопрос/ответ. Допустим, вы ввели в строке своего браузера ipipe.ru. Рассмотрим работу DNS пошагово:

Шаг 1. Ваш браузер об IP адресе ipipe.ru ничего не знает и с запросом IP адреса ipipe.ru, через специальную программу resolver обращается к локальному серверу имен.

Локальный DNS сервер – это сервер имен вашей локальной сети или DNS сервер вашего Интернет провайдера. При настройках сетевого подключения вы прописываете IP адреса DNS серверов (или же этот параметр получен через DHCP) один из которых будет отвечать на запросы, посылаемые вашим браузером через resolver – это и есть локальный или местный сервер вашей сети. Если вы используете модемное подключение (через телефонную линию), то для Вас местным сервером имен – будет DNS сервер вашего провайдера. IP адрес этого сервера также будет прописан в настройках сетевого подключения, не зависимо от того как осуществлялась настройка (вручную или автоматически). Вы всегда сможете посмотреть IP-адрес вашего локального DNS сервера.

Шаг 2. Запрос на IP адрес ipipe.ru доходит до местного сервера имен. Этот сервер об этом IP адресе ничего не знает, и посылает запрос одному из корневых серверов «.» (root).

Шаг 3. Корневой сервер отдает локальному серверу IP адрес сервера, который поддерживает зону .ru.

Шаг 4. Далее по полученному адресу локальный сервер имен обращается к DNS серверу, который поддерживает .ru.

Шаг 5. Этот DNS сервер по полученному запросу отдает IP адрес сервера, который поддерживает зону ipipe.ru.

Шаг 6. Местный DNS сервер с запросом IP адреса ipipe.ru обращается к DNS серверу зоны ipipe.ru.

Шаг 7. Локальный сервер имен получает IP адрес ipipe.ru. от DNS сервера зоны ipipe.ru.

Шаг 8. Получив адрес ipipe.ru локальный DNS сервер сообщает его Вашему браузеру.

Теперь браузер знает IP адрес ipipe.ru и напрямую обращается к серверу, на котором расположен сайт ipipe.ru.

Простейший алгоритм работы представлен на рисунке 2.

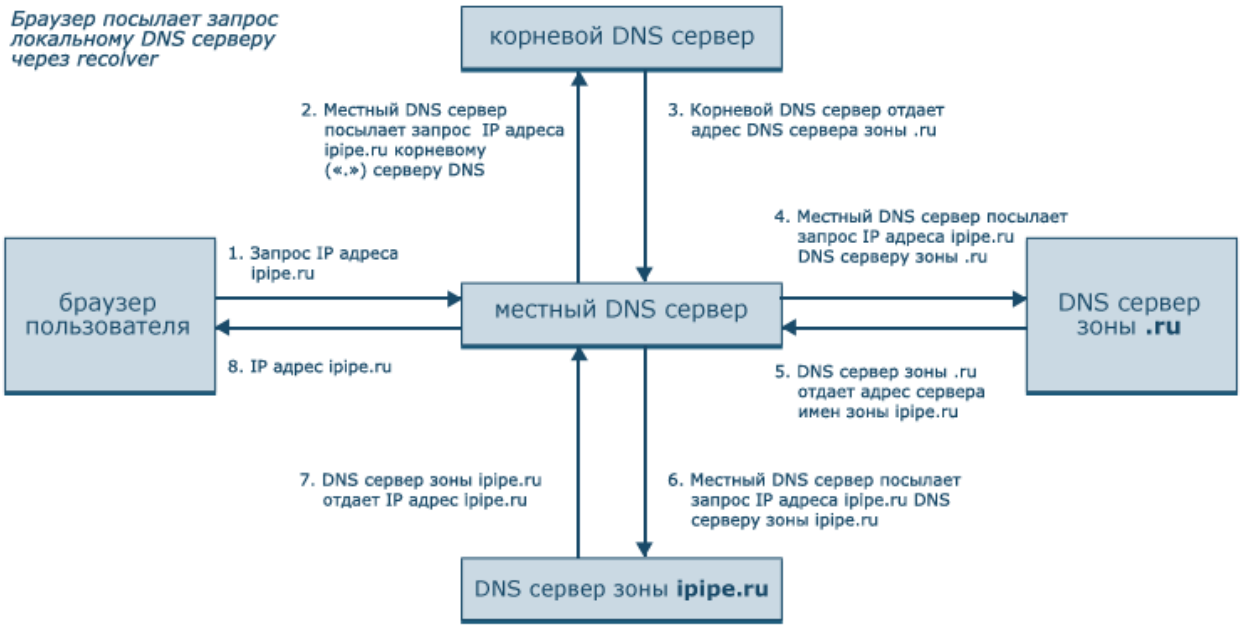

Рисунок 2 – Алгоритм работы получения IP-адреса по имени

Пространство доменных имен (информация о соответствии имен узлов IP-адресам) организовано иерархически в так называемые зоны. Понятие зоны должно быть уже знакомо вам по работе в Интернете. Иногда вместо слова «зона» говорят «домен», но это не совсем точно: зона может включать пространство нескольких доменов.

Локальная сеть образует одну зону. Нужно выбрать для нее название. Если в ваши намерения не входит «публикация» имен узлов вашей сети в Интернете, то название зоны можете выбирать произвольно.

DNS определяет:

– иерархическое пространство имен компьютеров и IP-адресов;

– таблицу имен компьютеров и адресов, реализованную в виде распределённой базы данных;

– «распознаватель» – клиентскую библиотеку функций, осуществляющих запросы к базе данных DNS;

– усовершенствованные средства маршрутизации электронной почты;

– механизм поиска служб в сети;

– протокол обмена информацией об именах.

Пакет BIND

Существует несколько программных реализаций сервера DNS. Наиболее распространенной реализацией является BIND. Демон ***named*** собственно реализует функцию разрешения имен на IP-адреса. Если ему неизвестно какое-либо имя, то он осуществляет разрешение этого имени при помощи других серверов. Серверы имен могут быть нескольких типов (Таблица 1 – Типы серверов DNS) и обычно один сервер совмещает сразу несколько режимов работы.

Таблица 1 – Типы серверов DNS

Тип сервера	**Описание**
Авторитарный	Официальный представитель зоны, ответы этого сервера самые точные и не устаревшие.
Первичный	Основное хранилище данных зоны. Вся информация хранится здесь.
Вторичный	Копирует данные с главного сервера и страхует главный сервер.
Неавторитарный	Отвечает на запросы из своего кэша. В принципе, ответ может быть некорректным, хотя такое встречается редко.
Рекурсивный	Осуществляет запросы от имени клиента до полного разрешения адреса. Клиенту выдается соответствие «имя – адрес».
Нерекурсивный	Перенаправляет клиента к другому серверу. Как правило, все серверы верхнего уровня нерекурсивные, так как они обрабатывают большое число запросов.

Основная настройка пакета BIND производится путём редактирования файла **/etc/bind/named.conf**. В данном файле по умолчанию указаны ссылки на файлы с описанием зон:

include "/etc/bind/named.conf.options"; – файл с опциями.

include "/etc/bind/named.conf.local"; – файл для описания зон (пустой).

include "/etc/bind/named.conf.default-zones"; – файл с зонами по умолчанию.

В файле с зонами по умолчанию описаны 5 зон (**должны всегда присутствовать**):

«.», «localhost», «127.in-addr.arpa», «0.in-addr.arpa», «255.in-addr.arpa».

Зоны описываются следующей конструкцией:

zone "*имя_зоны****" {***

type тип_зоны;

file «имя файла с полным путём»; – только для основной и дочерней зоны;

allow-update{ список хостов и сетей;}; – в случае динамического обновления;

allow-query{ список хостов и сетей;}; – разрешение ответа из сетей;

forwarders {список dns-серверов;}; – только для зоны пересылки;

forward only или first; – first; действует только при непустом списке forwarders; перенаправлять запросы, не имеющие ответов в кэше или своих зонах, серверам, указанным в списке forwarders; позволяет организовать общий кэш для нескольких серверов или доступ в Интернет через прокси; first – сначала делается запрос к серверам из списка, при неудаче производится собственный поиск; only – собственный поиск не производится;

зоны

};

тип_зоны может принимать значения: ***master*** (основная зона), ***slave*** (вторичная зона), ***forward*** (для зоны пересылки).

Список адресов может быть представлен следующими вариациями: ***any***, ***none***, *адрес*, *адрес_подсети/маска_подсети* (количество единиц в сетевой маске).

Для добавления зоны требуется создать запись о зоне в файле ***named.conf.local*** (либо создать новый файл и подключить его), описать зону в отдельном файле и выполнить перезапуск службы.

Файл ***/etc/bind/named.conf*** ссылается на файл ***/etc/bind/named.conf.options***, в котором содержится параметры для настройки BIND. Внутри конструкции options задаются определённые параметры. Нас будут интересовать listen-on (список адресов интерфейсов, которые могут принимать запросы), и allow-recursion (список хостов и сетей, от имени которых выполняются рекурсивные запросы).

Примеры параметров:

listen-on {127.0.0.1;192.168.2.2;!192.168.2.1;}; (разрешить принимать запросы с внутреннего интерфейса и с 192.168.2.2, запретить с 192.168.2.1);

allow-recursion {127.0.0.1;192.168.156.192/27;};

Настройка базы данных DNS

Директивы в файлах зон

Доменная база представляет собой набор текстовых файлов, которые правит администратор. В них содержаться записи двух типов: директивы синтаксического анализаторы (***$ORIGIN***, ***$TTL*** и др.) и записи о ресурсах.

Директивы синтаксического анализатора служат только для облегчения ввода данных. Директивы должны стоять в первой колонке и занимать отдельную строку.

Директива ***$ORIGIN*** позволяет сменить используемое имя домена. Имеет вид: **$ORIGIN** *имя_домена*. **Когда сервер читает файл зоны, он**

добавляет стандартное имя домена к любому имени, которое полностью не определено (т.е. не заканчивается на «.»). Источником имени зоны первоначально служит имя домена, указанного в инструкции ***zone***. Это можно исправить использую данную директиву. В случае указания нескольких директив, используется последняя встретившиеся.

Директива ***$TTL*** задает стандартное значение для поля ***ttl*** последующих записей. Данная директива должна предшествовать записи SOA. По умолчанию используются секунды, но можно указать и часы(h), минуты(m), дни(d), недели(w). Например, *$TTL 24h ;24 часа.*

Директива ***$TTL*** определяет, как долго запись о сопоставлении доменного имени IP-адресу будет храниться в кэше сервера.

Спецсимволы в файлах зон

Все параметры могут быть разделены табуляцией, пробелами и спецсимволами. Спецсимволы представлены в Таблица 2 – Специальные символы, используемые в записях о ресурсах.

Таблица 2 – Специальные символы, используемые в записях о ресурсах

Символ	Назначение
;	Начало комментария
@	Имя текущей зоны
()	Разбивка данных на несколько строк
*	Метасимвол

Записи о ресурсах в файлах зон

Существует различные типы DNS-записей:

– зонные записи – определяют домены и их серверы имён;

– базовые записи – связывают имена с адресами и обеспечивают маршрутизацию электронной почты;

– аутентификационные записи – предоставляют информацию, касающуюся аутентификации и сигнатур;

– вспомогательные записи – содержат дополнительную информацию о компьютерах и доменах.

В Таблица 3 – Записи базы данных DNS представлены записи базы данных DNS.

Таблица 3 – Записи базы данных DNS

	Тип	Название	Назначение/содержимое
Базо Зонные	SOA	Start of Authority – начало полномочий	Определение DNS-зоны
	NS	Name Server – сервер имен	Определение серверов имен зоны, делегирование полномочий поддоменам
	A	IPv4 Address – адрес IPv4	Преобразование имени в IPv4 адрес

	Тип	Название	Назначение/содержимое
	AAAA	IPv6 Address – адрес IPv6	Преобразование имени в IPv6 адрес
	PTR	Pointer – указатель	Преобразование адреса в имя
	MX	Mail Exchanger – обмен почтой	Маршрутизация электронной почты
Аутентификационные	DS	Delegation Signer – подписывающая сторона, выполняющая делегирование	Хэш ключа подписания подписанной дочерней зоны
	DNSKEY	Public Key – Открытый ключ	Открытый ключ для DNS-имени
	NSEC	Next Secure – Следующая защита	Используется наряду с DNSSEC для отрицательных ответов
	RRSIG	Signature – Сигнатура	Набор подписанных аутентифицированных записей о ресурсах
Вспомогательные	CNAME	Canonical Name – Каноническое имя	Дополнительные имена(псевдонимы) хоста
	LOC	Location – месторасположение	Географическое месторасположение и физический размер DNS-объекта
	SRV	Services – службы	Месторасположение известных служб в пределах домена
	TXT	Text – текст	Комментарии или нетипизированная информация

Запись SOA

Формат записи SOA можно представить, как:

*zone ttl **IN SOA** origin contact (serial refresh retry expire minimum)*

В этой записи каждое из полей обозначает следующее:

Поле **zone** - имя зоны. В случае использования вместо имени зоны спецсимвола «@», будет использовано имя зоны из описания в файле ***named.conf***. Поле зоны обязательно должно быть указано, иначе named не сможет привязать следующие за данной записью описания ресурсов к имени зоны. Можно, конечно, указывать и полное имя зоны, не забывая при этом ставить на конце имени «.»:

kyky.ru. IN SOA ns.kyky.ru adm.kyky.ru (2 8h 30m 2w 2h)

Поле **ttl** в записи SOA всегда пустое. Дело в том, что время кэширования для записей описания зоны задается либо последним аргументом данных записи SOA, либо директивой управления **$TTL.**

Поле **origin** – это доменное имя основного сервера зоны. В случае описания зоны kyky.ru в качестве сервера используется машина ns.kyky.ru. Данное доменное имя и должно быть указано в поле **origin.**

Поле **contact** определяет почтовый адрес лица, осуществляющего администрирование зоны. Данный адрес должен совпадать со значением адреса указанным в заявке на домен. Есть, однако, одна особенность при указании этого адреса. Так как символ «@» имеет особый смысл при описании зоны, то вместо этого символа в почтовом адресе используется символ «.» (точка). Например, если ваш администратор домена имеет почтовый адрес adm@kyky.ru, то в поле **contact** следует писать не adm@kyky.ru, а adm.kyky.ru.

Поле данных в записи SOA разбито на аргументы, которые определяют порядок работы сервера с записями описания зоны. Как правило, все аргументы располагают на другой строке или, для лучшего отображения, каждый на своей строке, что заставляет записывать их внутри скобок.

Атрибут **serial** – определяет серийный номер файла зоны. Если говорить проще, то в этом поле ведется учет изменений файла описания зоны. В принципе это могут быть любые числа, но чаще всего администраторы используют в качестве серийного номера год (4 позиции), месяц (две позиции), день (две позиции) и версию внесения изменений в файл описания зоны (две позиции). Важность серийного номера определяется тем, что когда вторичный (secondary) сервер обращается к первичному (primary) серверу для обновления информации о зоне, то он сравнивает серийный номер из своего кэша с серийным номером из базы данных первичного сервера. Если серийный номер из primary сервер больше, то secondary сервер обращается к primary и копирует описание всей зоны целиком, если нет, то он не вносит изменений в свою базу данных.

Атрибут **refresh** определяет интервал времени, после которого slave сервер обязан обратиться к master серверу с запросом на верификацию своего описания зоны, при этом проверяется серийный номер описания зоны.

Атрибут **retry** начинает играть роль тогда, когда основной сервер по какой-либо причине не способен удовлетворить запрос вторичного сервера за время, определенное атрибутом refresh. А если говорить точнее, то в момент наступления времени синхронизации описания зоны основной сервер по какой-либо причине не отвечает на запросы вторичного сервера.

Атрибут **expire** определяет интервал времени, после которого вторичный должен прекратить обслуживание запросов к зоне, если он не смог в течение этого времени верифицировать описание зоны, используя информацию с основного сервера.

Запись NS

Запись NS идентифицирует серверы имен, являющиеся авторитетными для данной зоны (т.е. все главные и подчинённые сервера). Обычно эти записи стоят после записи SOA. Формат записи таков:

имя_зоны ***IN NS*** *имя_хоста*

Пример:

stuff.org. IN NS first.stuff.org.

stuff.org. IN NS second.stuff.org.

Здесь для домена stuff.org. задано два сервера имен: first.stuff.org. и second.stuff.org.

Если имя зоны совпадает с именем зоны в записи **SOA**, то поле *имя_зоны* можно пропустить.

Следует указывать все авторитетные сервера имен домена, которые должны быть видны внешнему миру, причем это необходимо сделать как в зонном файле домена, так и в зонном файле родительского домена.

Запись A

Данная запись используется **только** в прямых зонах.

Эта запись является основной записью базы данных DNS и отображает имя в адрес. Для каждого из сетевых интерфейсов обычно существует своя запись A. Для компьютера может существовать несколько записей, по каждой на один сетевой интерфейс. Формат записи такой:

имя_хоста ***IN A*** *ip_адресс*

Пример:

first.stuff.org. ***IN A*** *128.100.241.212*

first ***IN A*** *128.100.241.213*

second.stuff.org. ***IN A*** *128.100.241.214*

Здесь первые две записи относятся к одному компьютеру с именем first.stuff.org. и двумя сетевыми картами, имеющими IP-адреса 128.100.241.212 и 128.100.241.213.

Третья запись относится ко второму компьютеру second.stuff.org., сетевая карта которого имеет адрес 128.100.241.214.

Запись PTR

Данная запись используется в обратных зонах.

Эта запись обеспечивает обратный перевод, то есть перевод IP-адресов в символьные имена. Точно так же, как и в предыдущем случае, для каждого сетевого интерфейса существует только одна запись PTR. В простейшем случае это записанный наоборот IP-адрес *(смотрите пример записи A)* с суффиксом in-addr.arpa.

Пример:

212 IN PTR first.stuff.org.; в данном случае будет использоваться имя зоны

213.241.100.128.in-addr.arpa. IN PTR first.stuff.org.; явное указание адреса

214.241.100.128.in-addr.arpa. IN PTR second.stuff.org.

Пример файла прямой зоны:

$TTL 6800

$ORIGIN .

study.local IN SOA ubuntu.study.local. mail.study.local. (

1

7200

1800
604800
7200
)
$ORIGIN study.local
@ NS ubuntu.sudy.local.
ubuntu A 192.168.1.1
ubuntu.study.local. A 192.168.2.1

Ход работы

Для установки службы DNS сервера bind9 вызывается команда: ***apt-get install bind9.***

Запуск службы осуществляется командой: ***/etc/init.d/bind9 start.***

Рестарт службы осуществляется командой: ***/etc/init.d/bind9 restart.***

Остановка службы осуществляется командой: ***/etc/init.d/bind9 stop.***

Для возможности обновления файлов зоны DNS, в конфигурационном файле службы DHCP необходимо для параметра ***ddns-update-style*** выставить параметр ***interim***.

Для получения полного доступа к файлу службой BIND необходимо выполнить команду ***chown bind:bind имя_файла***

Файлы обновляемых зон DNS рекомендуется размещать в папке ***/var/lib/bind***.

Для организации логирования BIND, необходимо в файле ***/etc/rsyslog.d/50-default.conf*** добавить запись вида:

!named

**.* /var/log/имя_файла_лога*

Перезапуск системы логирования осуществляется командой ***service rsyslog restart***

Для проверки конфигурации служат следующие команды:

- named-checkconf;
- named-checkzone.

Описания команд можно просмотреть с помощью команды man.

При корректной работе будут созданы файлы журнала в папке рядом с файлами зон с расширением jnl.

Практическая часть

Содержание задания

В рамках задания требуется установить и настроить DNS-сервер для Ubuntu Server. По результатам работы составить отчет.

Порядок выполнения работы

1 Ознакомиться с теоретической частью.

2 Установить Ubuntu Server.

3 Установить и настроить DHCP-сервер.

4 Установить и настроить DNS-сервер (обязательно наличие прямой и обратной зон). **Имя домена:** *Фамилия_студента.local*. Прямая и обратная зона должны обновляться при подключении клиентов.

5 Проверить наличие созданных BIND *.jnl – файлов.

6 Проверить наличие автоматически созданных записей в файлах зон.

7 Создать зону пересылки так, чтобы клиент имел возможность узнать ip-адрес клиента из зоны пересылки.

8 Настроить дочерний домен.

9 Составить отчёт о проделанной работе.

10 Показать выполненную работу и отчёт преподавателю.

Содержание отчёта

1 Титульный лист.

2 Цель работы.

3 Краткие теоретические сведения.

4 Основные результаты по каждому пункту хода выполнения работы.

5 Описание проверок функционирования служб со скриншотами результатов.

6 Выводы о проделанной работе.

Защита работ проводится индивидуально.