Лабораторная работа №5
Система доменных имен

Введение.

Каждому компьютеру, расположенному в локальной сети возможно назначить свой IP адрес – число, по которому к нему возможно будет обратиться с другого устройства в сети. Однако, для удобства использования каждому числовому адресу возможно сопоставить символьное значение. Представьте телефонную книгу, содержащую номера и имена их владельцев. Человеку гораздо проще запомнить имя и найти по нему номер, чем держать в голову миллионы бессвязных чисел. Отсюда возникает проблема организации такого телефонного справочника - системы для сопоставления адреса и имени устройства.

Изначально, в сети ARPAnet – предке современного интернета число узлов составляло несколько сотен. Поэтому проблема преобразования численного IP адреса в символьное имя решалось следующим способом: всю необходимую информацию содержал файл HOSTS.TXT, который находился на каждом из компьютеров в сети и редактировался с появлением новых устройств администраторами сетевого информационного центра (NIC, Network Information Center), расположенного в Северной Америке. Знакомый вам файл в Unix-системах /etc/hosts унаследовал его структуру.

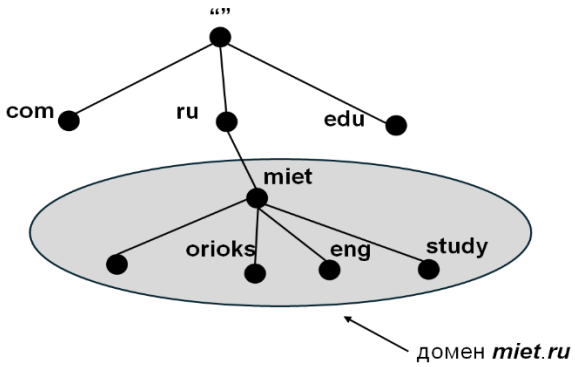
Однако, когда число узлов в сети стало резко расти, то возникли следующие проблемы:

1. Выросла нагрузка на сервера Сетевого информационного центра – при добавлении нового узла в сеть, на каждый существующий узел необходимо отправить обновление
2. Из-за огромного числа имен в файле HOSTS.TXT стали допускаться ошибки и их повторения, что приводило к конфликтам
3. Необходимость выполнение синхронизации с большей частотой – пока обновление достигало восточного берега США – уже появлялись новые адреса.

Все перечисленные проблемы привели к тому, что стало необходимо систематизировать имена компьютеров и разработать специальную базу данных, содержащих сопоставление имен и адресов. Каждый компьютер, зная адрес базы мог к ней обратиться и определить IP устройства по имени. Такая распределенная база данных называется DNS.

Система доменных имен.

DNS *(Domain Name System)* — это распределенная база данных, которая содержит информацию о компьютерах, включенных в сеть Internet [1]. База данных представлена в виде перевернутого дерева, похожего на файловую систему UNIX. В качестве корневого узла выступает пустой символ «», от которого идут ветви, также оканчивающиеся узлами. Каждый из них является корнем для новой ветви дерева. Такие ветви являются разделом базы данных и называются **доменом**. Домены включают в себя узлы и **поддомены** – участки домена.



Например, на рисунке *miet.ru* является доменом. Для него orioks.miet.ru – поддомен.

Современные DNS организованы по принципу клиент-сервер, где каждое устройство, желающее получить значение IP адреса по доменному имени обращаются к DNS серверу, на котором расположена сама база или адрес другого сервера, владеющего этой информацией. Проблема заключается в том, что если хранить все сопоставления на одном, пускай и очень мощном сервере, то он всё равно не справится с нагрузкой всей сети и поиск сопоставления имени составит большое число времени. Поэтому было предложено решение, когда имена каждого из узлов в сети представляют собой набор из символьных меток, разделенных символом «.». Каждая метка закреплена за определенным узлом, расположенном на сервере. Поиск имени происходит с самой верхней метки, которая содержит адреса серверов меток следующего уровня и оканчивается на сервере, хранящим значение имени узла.

Например, узел orioks.miet.ru. – содержит в себе три символьные метки – orioks, miet, ru. Для поиска его IP адреса, изначально будет проведено обращение на самый верхний, корневой сервер, на котором хранятся адреса узлов com, ru, miet. Найдя адрес сервера с данными узла ru, происходит обращение к нему по найденному адресу. Далее аналогично по шагам вычисляются адреса miet и orioks. Имя orioks.miet.ru. – является **полным доменным именем**.

Для того, чтобы упростить работу, дерево DNS было разделено на отдельные **зоны**, которые администрируются независимо друг от друга. Возможно произвести настройку прямой и обратные зоны. Прямая зона отвечает за преобразование из доменного имени в IP адрес, обратная – наоборот, за преобразования из IP адреса в доменное имя.

Настройка DNS сервера

Существует несколько реализаций DNS-серверов. К ним относят *bind9*, *PowerDNS, Dnsmasq, djbdns* и некоторые другие. Каждая из них обладает своими особенностями. Например, *Dnsmasq* легковесный dns-сервер, который возможно использовать в локальной сети до 1000 клиентов. Он устанавливается по умолчанию на ОС Astra Linux. У *PowerDNS* открытый исходный код и он часто используется для организации балансировки DNS-трафика ряда крупных веб-сайтов. Djbdns – набор утилит для обслуживания DNS, отличается высокой безопасностью и надежностью.

Однако, далее будет рассмотрен пример организации DNS-сервера на основе bind9 – стандартной и классической утилиты, являющейся одной из самых популярных среди аналогов.

bind9

BIND (Berkeley Internet Name Domain) — программа, реализующая функции DNS-сервера. Является одной из самых популярных и распространенных. Установка продукта может быть произведена с помощью команды:

```
sudo apt install bind9
```

Настройка производится с помощью редактирование системных файлов, расположенных в директории /etc/bind.

Путь к файлу	Назначение
/etc/bind/named.conf	Основной конфигурационный файл. Содержаться директивы include и править его не нужно
/etc/bind/named.conf.options	Файл конфигурации, содержащий описание глобальных параметров
/etc/bind/named.conf.local	Файл конфигурации, содержащий описание зон
/etc/bind/named.conf.default-zones	Файл конфигурации зон "по умолчанию"

Для описания глобальных параметров DNS сервера используется файл named.conf.options. Он содержит в себе глобальную секцию options, в которую необходимо вносить секции с настройками. В общем виде параметры файла указаны ниже:

```
тип_секции [имя_секции] {
    установки;
    установки;
    …
};
```

Перечислим некоторые типы секций:

Название секции	Описание	Пример
directory	Указывает каталог, где BIND будет искать файлы зон и другие файлы	directory "/var/cache/bind";

forwarders {список адресов DNS серверов;}	Определяет список DNS-серверов, к которым BIND будет перенаправлять запросы, если он не может их обработать локально	forwarders {
		8.8.8.8;
		8.8.4.4;
		};

listen-on{
интерфейсы;
}
 	
listen-on-v6 {
интерфейсы IPv6;}
	Задает IP-адреса и порты, на которых сервер BIND будет прослушивать запросы	listen-on {
		127.0.0.1;
		192.168.1.1;
		};
		listen-on-v6 {
		any;
		};

dnssec-validation	Управляет проверкой DNSSEC для зон	dnssec-validation auto;
allow query	Задает список IP-адресов или сетей, которым разрешено отправлять DNS-запросы	allow-query { any; };
querylog	Включает или отключает ведение журнала запросов	querylog yes;

Пример файла /etc/bind/named.conf.options

```
options {
    directory "/var/cache/bind";
    forwarders { 8.8.8.8; 8.8.4.4; };
    allow-query { any; };
    dnssec-validation auto;
    listen-on { 127.0.0.1; };
    listen-on-v6 { none; };
    querylog yes;
};
```

Для описания доменных зон, которые будут обслуживаться сервером необходимо внести изменения в файл /etc/bind/named.conf.local. Формат записи параметров файла идентичен файлу named.conf.options.

Поле тип секции может принимать различные значения, однако нас интересует только один тип – **zone.** Секция zone описывает одну конкретную доменную зону.

Установки секций также очень разнообразны – остановимся только на наиболее важных для нас.

Название секции	Описание	Пример
type	Определяет тип зоны: master/slave	type master
file	Указывает путь к файлу зоны, где хранятся записи для данной зоны	file "/etc/bind/db.example.com";
masters	Указывает адрес главного сервера на подчиненном	masters {192.168.122.2;};
allow-transfer	Указывает, каким серверам разрешено получать копию зоны. Это обычно используется для вторичных серверов	allow-transfer { 192.168.1.2; };
allow-update	Указывает, каким клиентам или серверам разрешено вносить изменения в записи DNS для определенной зоны в режиме реального времени	allow-update { 192.168.1.100; }; allow-update { key rndc-key; };
notify	Определяет, будут ли вторичные серверы уведомляться о изменениях в зоне	notify yes;

Поле type может содержать два значения – master и slave. DNS сервера могут работать в одном из двух режимов. При большой нагрузке полезно разделить DNS-сервер на несколько устройств – главный (master) хранит всю информацию о зонах, которую на нём возможно изменять. Ведомых (slave) серверов может быть несколько, они принимают запросы от клиентов и обрабатывают их. Необходимые значения получаются от master сервера и хранится в памяти ведомого.

Для работы трансфера зоны необходимо настроить два DNS сервера – master и slave. Для этого в параметрах файла named.conf.local в поле type на каждом из серверов необходимо указать его тип. На подчиненном сервере в описании зоны следует добавить строку masters с указанием адреса главного сервера.

Секция файл содержит имя файла, содержащее описание доменной зоны. Для её описания служит файл ресурсных записей. В общем виде файл возможно представить следующим образом:

ВРЕМЯ ЖИЗНИ

ИМЯ КЛАСС ТИП ДАННЫЕ

ВРЕМЯ ЖИЗНИ - $TTL (Например, $TTL 604800)

ИМЯ – имя ресурсной записи (Например, miet.stu)

КЛАСС – IN (от INternet)

ТИП – (SOA, A, AAAA, PTR, NS, MX, CNAME, SRV)

ДАННЫЕ – могут состоять из нескольких полей

Обычно первой строчкой описывается время, на которое запись о доменном имени считается действительной. Чтобы не обращаться к серверу много раз, полученные записи хранятся у клиента в кэше в течение времени TTL. Значение параметра указывается в секундах.

Следующие строки файла записываются в соответствии с шаблоном, указанным выше. Первой строкой обычно является запись типа SOA (Start of Authority) -

показывает, какой DNS сервер является ведущим для данной зоны, и определяет основные параметры для неё.

Для SOA записи поле с данными принимает следующие значения:

- Первое поле данных. - FQDN ведущего (master) сервера DNS для данной зоны.
- Второе поле данных определяет почтовый адрес администратора, ответственного за поддержку ведущего (master) сервера.
- Третье поле данных - последовательный номер (serial number), который определяет версию ресурсных записей данной зоны. Этот номер должен увеличиваться при каждом изменении данных о зоне для информирования подчиненных (slave) серверов о произошедших изменениях. Формат – yyyymmddvv.
- Четвертое поле данных - время обновления, то есть временной интервал, через который подчиненные (slave) серверы должны опрашивать ведущий (master) сервер, не изменились ли ресурсные записи зоны.
- Пятое поле предназначено для задания интервала времени, через которое подчиненный сервер повторит попытку обновления информации о зоне, если первая попытка обновления была неудачной.
- Шестое поле данных задает временной интервал, через который подчиненный сервер, не добившись связи с мастером, прекратит поддержку данной зоны.
- Седьмое поле данных определяет время жизни данных кэширования отрицательных ответов DNS сервера.

Записи **NS** следуют после записей SOA в файлах описания зон и предназначены для указания всех авторитетных (уполномоченных) серверов для данной зоны.

Записи типа **A** устанавливают соответствие между доменными именами хостов и IPv4 адресами хостов.

Записи типа **AAAA** устанавливают соответствие между доменными именами хостов и IPv6 адресами хостов.

CNAME - В первом поле задается альтернативное доменное имя (псевдоним), в последнем поле доменное имя хоста.

Для обеспечения обратного отображения IP адресов в имена хостов предназначены **PTR** записи

MX запись предназначена для указания, на какой хост должна быть отправлена почта вместо заданного в почтовом адресе хоста (или домена).

SRV записи предназначены для распределения нагрузки и создания резервных служб (расширение MX записей).

Для проверки настроек файлов bind служат специальные утилиты, устанавливаемые вместе с пакетом.

Параметры конфигурации возможно проверить с помощью команды:

```
named-checkconf /etc/bind/named.conf
```

Корректность настройки доменных зон можно проверить командой

```
named-checkzone имя доменной зоны
```

Обратите внимание, что при настройке подчиненного DNS сервера создавать файлы описания зон не требуется. Они будут переданы автоматически с основного сервера.

Динамический DNS

При настройке DNS сервера за каждым устройством будет закреплен определенный IP адрес. Изменение этого адреса приводит к необходимости изменения в файле описания зоны. Если IP адреса выдаются динамически, с помощью DHCP это приведет к постоянному обновлению конфигурации DNS администратором. Чтобы упростить задачу, возможно настроить динамический DNS.

Динамический DNS (Dynamic DNS, DDNS) — это служба, позволяющая автоматически обновлять запись доменного имени при изменении IP-адреса устройства. Когда IP-адрес устройства меняется (например, из-за перезагрузки роутера), DDNS автоматически обновляет запись в DNS-сервере, чтобы доменное имя продолжало указывать на правильный IP-адрес. Это особенно полезно для удаленного доступа к устройствам или серверам, у которых нет постоянного статического IP-адреса.

Настройка DDNS происходит следующим образом:

1. Создается симметричный ключ для шифрования данных обновлений
2. Настройка DNS. В файле named.conf.local в секцию zone добавляется директива allow-update с указанием имени ключа
3. Настройка DHCP. В файле dhcpd.conf указываются следующие параметры:
    a. Поддержка DDNS

```
ddns-update-style standard;
ddns-updates on;
```

ddns-update-style [ad-hoc | standard | interim | none] – параметр конфигурационного файла dhcpd.conf задается стиль обновлений DNS, которые будет выполнять DHCP-сервер.

    b. Имя зоны

```
option domain-name “имя зоны”
option domain-search “имя зоны”
```

    c. Указание ключей для зон DNS

zone «имя зоны» {primary «IP DNS сервера»; key «имя ключа»};

```
zone "mpsu.stu" {
    primary 192.168.122.1;
    key “rndc-key”;
};
```

d. Дополнительные параметры:

```
update-static-leases on;
do-forward-updates on;
update-conflict-detection false;
update-optimization false;
allow client-updates;
```

update-static-leases [on | off]: включает обновление статических записей для устройств, которым назначены статические IP-адреса.

do-forward-updates [on | off]: включает обновление прямых DNS-записей (A-записей) при назначении или освобождении IP-адресов

update-conflict-detection [true | false]: управляет поведением DHCP-сервера при обнаружении конфликта при обновлении DNS-записей

- **true** (по умолчанию): включает обнаружение конфликтов при обновлении DNS. Если сервер находит существующую запись для имени, которое он пытается обновить, он не будет перезаписывать эту запись.
- **false**: отключает обнаружение конфликтов, и сервер будет обновлять DNS-записи, даже если существует запись для того же имени.

update-optimization [true | false]: управляет оптимизацией обновлений DNS-записей

- **true:** (по умолчанию): включает оптимизацию обновлений. DHCP-сервер обновляет записи DNS только тогда, когда IP-адрес или имя хоста изменяется.
- **false**: отключает оптимизацию обновлений, и сервер будет обновлять DNS-записи каждый раз, когда клиент получает новый или продлевает существующий арендуемый IP-адрес, независимо от того, изменился ли IP-адрес или имя хоста.

[allow | deny] client-updates: управляет разрешением клиентам самостоятельно обновлять свои DNS-записи

- **allow**: позволяет клиентам выполнять собственные обновления DNS-записей.
- **deny**: запрещает клиентам самостоятельно обновлять свои DNS-записи. В этом случае DHCP-сервер берет на себя управление обновлениями DNS.

Пример настройки DNS сервера.

Приведем пример настройки доменной зоны mpsu.stu. Для неё будет настроена прямая и обратная зоны.

Файл named.conf.local

```
zone "mpsu.stu" {
    type master;
    file "/etc/bind/zones/miet.stu";
};

zone "122.168.192.in-addr.arpa" {
    type master;
    file "/etc/bind/zones/192.168.122.stu";
};
```

Файл ресурсной записи для прямой зоны

```
$TTL 604800
mpsu.stu.           IN      SOA srv.mpsu.stu. admin@mpsu.stu  (
                                2024030901   ;Последовательный номер
                                3h     ;Обновление
                                1h     ;Повтор попытки обновления
                                1w     ;Устаревание через 1 неделю
                                1h     ;TTL отрицательного кэширования
                                )
mpsu.stu.           IN      NS          srv.mpsu.stu.
srv.mpsu.stu.       IN      A           192.168.122.2
cli.mpsu.stu.       IN      A           192.168.122.3
neighbor            IN      CNAME   cli.mpsu.stu.
```

Файл ресурсной записи для обратной зоны

```
$TTL 604800
122.168.192.in-addr.arpa.       IN      SOA srv.mpsu.stu. admin@mpsu.stu        (

                                        2024030901   ;Последовательный номер
                                        3h           ;Обновление
                                        1h           ;Повтор попытки обновления
                                        1w           ;Устаревание через 1 неделю
                                        1h           ;TTL отрицательного кэширования
                                                        )
122.168.192.in-addr.arpa.       IN      NS      srv.mpsu.stu.

2                       IN      PTR     srv.mpsu.stu.
3                       IN      PTR     cli.mpsu.stu.
```

Обратите внимание! С символа «точка с запятой» в конфигурационных файлах начинаются комментарии. Закомментировать строку с помощью символа # не получится.

Проверка настройки осуществляется следующим образом:

```
named-checkconf /etc/bind/named.conf
named-checkzone mpsu.stu /etc/bind/zones/mpsu.stu
named-checkzone 122.168.192.in-addr.arpa /etc/bind/zones/122.168.192.stu
```

Если всё успешно, то перезапустим bind9

```
sudo systemctl restart bind9
```

Чтобы на машине клиента указать адрес DNS сервера, необходимо его добавить в файле resolv.conf. Для указания домена используйте ключевое слово domain.

При отправке ping сообщения по доменному имени должно произойти обращение к доменному серверу и преобразование имен.

Задание 1.

1. На виртуальные машины server и node1 установите пакет bind9 (или убедитесь, что он уже установлен)
2. На server машине проделайте следующие шаги:
    1. Опишите зону DNS «Ваши_инициалы.miet.stu» (Например, pmn.miet.stu)
    2. Опишите обратную зону DNS для подсети 192.168.122.0
    3. Проверьте правильность внесенных изменений
    4. Создайте каталог /etc/bind/zones. В нем создайте файлы с ресурсными записями для созданных вами зон. Включите в данную зону четыре машины – три созданные вами (server, node1,node2) и еще одну с именем node2 и адресом 192.168.122.(N в группе + 3)
    5. Проверьте правильность внесенных изменений
    6. Перезапустите bind9 и поочередно отправьте ping сообщение машинам с именами server, node1, node2, node3. Объясните полученный результат
3. Настройте node1 и node2 таким образом, чтобы было возможно отправлять ping сообщения по доменным именам.

Список литературы

1. *Ли К., Альбитц П.* DNS и BIND, 5_е издание. – Пер. с англ. – СПб.: Символ_Плюс, 2008. – 712 с.
2. Курс AL-1704 Сетевое администрирование ОС Astra Linux Special Edition 1.7

3. *Немет Э., Хейн Т., Снайдер Г.* Unix и Linux: руководство системною администратора, 5-е изд.: Пер. с ан гл . - СПб. : ООО "Диалектика" , 2020. - 1168 с.