Bind 9 - настройка DNS-сервера

Опубликовано muff в Сб, 2009-08-22 12:34

По умолчанию во **FreeBSD** используется одна из версий программы **BIND** (**Berkeley Internet Name Domain**), являющейся самой распространенной реализацией протокола **DNS**.

FreeBSD в настоящее время поставляется с сервером DNS BIND9, предоставляющим расширенные настройки безопасности, новую схему расположения файлов конфигурации и автоматические настройки для **chroot**. **chroot-каталогом является /var/named, соответственно в конфигурационных файлах все пути будут относительно этого каталога.** Исходя из вышесказаного:

```
# cd /var/named
```

У нас уже есть пример конфигурационного файла. Сохраним его (на всякий случай), и создадим свой:

```
# mv etc/namedb/named.conf etc/namedb/named.conf.default
 # touch etc/namedb/named.conf
```

Конфигурирование сервера я начинаю с настройки утилиты управления rndc.

Воспользуемся утилитой **rndc-confgen** для генерации конфигурационного файла **rndc**:

```
# rndc-confgen >> etc/namedb/rndc.conf
```

Посмотрим результат:

```
# cat etc/namedb/rndc.conf


# Start of rndc.conf
key "rndc-key" {
algorithm hmac-md5;
secret "8moaOusPKbJDSaQjfvcrwA==";
};

options {
default-key "rndc-key";
default-server 127.0.0.1;
default-port 953;
};
# End of rndc.conf

# Use with the following in named.conf, adjusting the allow list as needed:
# key "rndc-key" {
# algorithm hmac-md5;
# secret "8moaOusPKbJDSaQjfvcrwA==";
# };
#
# controls {
# inet 127.0.0.1 port 953
# allow { 127.0.0.1; } keys { "rndc-key"; };
# };
# End of named.conf
```

Берем из полученного файла необходимые данные для **named.conf** (отмечено, что эти строки необходимо вставить в **named.conf**). Потом добавляем остальные опции:

```
# cat etc/namedb/named.conf


# rndc
key "rndc-key" {
algorithm hmac-md5;
secret "8moaOusPKbJDSaQjfvcrwA==";
};

controls {
inet 127.0.0.1 port 953
allow { 127.0.0.1; } keys { "rndc-key"; };
};
# end rndc

// Назначаем access-листы (потом пригодится...)

acl "client" { 127.0.0.1; 195.3.159.250/32; };
acl "slave" { x.x.x.x/x; y.y.y.y/y; };

// Настраиваем логирование. Нужная штука ;)

logging {
channel my-default {
file "/log/named" versions 5 size 10m;
print-time yes;
print-category yes;
print-severity yes;
};

channel general {
file "/log/general" versions 2 size 10m;
print-time yes;
print-category yes;
print-severity yes;
};

channel database {
file "/log/database" versions 2 size 10m;
print-time yes;
print-category yes;
print-severity yes;
};

channel security {
file "/log/security" versions 2 size 10m;
print-time yes;
print-category yes;
print-severity yes;
};

channel config {
file "/log/config" versions 2 size 10m;
print-time yes;
print-category yes;
print-severity yes;
```

```
};

channel resolver {
file "/log/resolver" versions 2 size 10m;
print-time yes;
print-category yes;
print-severity yes;
};

channel xfer-in {
file "/log/xfer-in" versions 2 size 10m;
print-time yes;
print-category yes;
print-severity yes;
};

channel xfer-out {
file "/log/xfer-out" versions 2 size 10m;
print-time yes;
print-category yes;
print-severity yes;
};

channel notify {
file "/log/notify" versions 2 size 10m;
print-time yes;
print-category yes;
print-severity yes;
};
channel client {
file "/log/client" versions 2 size 10m;
print-time yes;
print-category yes;
print-severity yes;
};

channel unmatched {
file "/log/unmatched" versions 2 size 10m;
print-time yes;
print-category yes;
print-severity yes;
};

channel network {
file "/log/network" versions 2 size 10m;
print-time yes;
print-category yes;
print-severity yes;
};

channel update {
file "/log/update" versions 2 size 10m;
print-time yes;
print-category yes;
print-severity yes;
};

channel queries {
file "/log/queries" versions 2 size 10m;
```

```
print-time yes;
print-category yes;
print-severity yes;
};

channel dispatch {
file "/log/dispatch" versions 2 size 10m;
print-time yes;
print-category yes;
print-severity yes;
};

channel dnssec {
file "/log/dnssec" versions 2 size 10m;
print-time yes;
print-category yes;
print-severity yes;
};

channel lame-servers {
file "/log/lame-servers" versions 2 size 10m;
print-time yes;
print-category yes;
print-severity yes;
};

category default { my-default; };
category general { my-default; general; };
category database { my-default; database; };
category security { my-default; security; };
category config { my-default; config; };
category resolver { my-default; resolver; };
category xfer-in { my-default; xfer-in; };
category xfer-out { my-default; xfer-out; };
category notify { my-default; notify;};
category client { my-default; client;};
category unmatched { my-default; };
category network { my-default; network; };
category update { my-default; update; };
category queries { queries; };
category dispatch { my-default; dispatch; };
category dnssec { my-default; dnssec; };
category lame-servers { lame-servers; };
};


// теперь пошли опции

options {

// Указываем рабочую директорию
directory "/etc/namedb";

// Прячем имя сервера (повышаем безопасность)
hostname "DNS server";

// Разрешаем рекурсивные запросы
recursion yes;
```

```
// Указываем путь к pid-файлу. Проверьте на всякий случай путь.

// Используется для остановки/перезапуска DNS-сервера BIND
pid-file "/var/run/named/pid";

// Указываем путь к дампу
dump-file "/var/dump/named_dump.db";

// Прячем версию BIND (повышаем безопасность)
version "Unknown DNS";

// Чьи запросы будем обрабатывать
allow-query { "client"; };

// Кому разрешаем трансфер наших зон

allow-transfer { "slave"; };

// Указывае размер кеша
max-cache-size 52428800 ;

};
### END options

// Корневая зона.
zone "." {
    type hint;
    file "/etc/namedb/named.root";
 };

// Back Resolving для нас loopback
zone "0.0.127.in-addr.arpa" {
        type master;
        file "/etc/namedb/localhost.rev";
        notify no;
};

// Подключаем дополнительные конфигурационные файлы
// Рекомендую для каждой зоны свой. Проще потом конфиг найти, и не захаращиваем
// основной конфиг всякой лабудой. Думаю здесь все понятно...

include "/conf/kiev.ua";

// В этот файл больше не ничего не писать. Используйте include . Заканчиваем пустой
строкой.
```

Разберем конфигурационный файл более детально.

Сначала идут опции для работы утилиты **rndc**. Потом секция настройки access-листов. Рекомендую использовать именно access-листы, а не прописывать всюду айпишки и сети - удобней будет ;)

Далее идет настройка логов. Даннная секция настроена так, чтобы разные действия писались в разные файлы. Разберетесь что и куда, если будет такая необходимость ;)

Кстати, нужно создать эту самую папку для логов...

```
# mkdir log
```

Далее идет секция опций. Думаю здесь все понятно.

Потом идет описание зон. С корневой зоной проблем не должно возникнуть, поскольку на данный момент используется тип slave, и список TLD-серверов должен подтягиваться непосредственно к нам с корневого ДНС-сервера F.

Рассмотрим файл бек-резолвинга для **loopback**. В дистрибутиве уже идет готовый файл, но я по привычке создаю свой (тем более IPv6 я пока не использую). Создадим файл **etc/namedb/localhost.rev**

```
# touch etc/namedb/localhost.rev
```

Приведем **localhost.rev** к следующему виду:

```
# cat etc/namedb/localhost.rev

$TTL 3600
@     IN     SOA      muff.kiev.ua.     admin.muff.kiev.ua. (
                                          2009082300 ; Serial
                                          3600 ; Refresh
                                          600 ; Retry
                                          3600000 ; Expire
                                          3600 ) ; Minimum

      IN     NS       muff.kiev.ua.
1     IN     PTR      localhost.muff.kiev.ua.
```

Рассматривать детально каждую опцию не буду. Этой информации достаточно в инете. Обращу внимание только на то, что все имена заканчиваются точкой, указывая что это уже корневой домен; запись **admin.muff.kiev.ua** - это почтовый адрес администратора домена, только "@" заменена на ".", поскольку "@" в даном случае имеет служебное значение.

Далее прописываю какие еще файлы инклюдить в основной конфиг. Чтобы не путаться, все зоны свожу в дополнительные конфиги в алфавитном порядке. Приведу пример только с одним дополнительным конфигом "kiev.ua" (потом обычно их становится намного больше: com, com.ua, net, net.ua и т.д.).

Создаем каталог для дополнительных конфигурационных файлов и файл конфигурации для доменов вида *.kiev.ua:

```
# mkdir conf
 # touch conf/kiev.ua
```

Теперь добавим поддержку домена **muff.kiev.ua**:

```
# cat conf/kiev.ua

zone "muff.kiev.ua" {
    type master;
    file "/zones/kiev.ua/muff.kiev.ua";
    allow-query {any ; };
    allow-transfer { "slave"; };
};
```

Создадим необходимые каталоги и файл зоны **muff.kiev.ua**.

```
# mkdir -p zones/kiev.ua
 # touch zones/kiev.ua/muff.kiev.ua
```

Добавим необходимые записи в файл зоны **muff.kiev.ua**. Должно получиться примерно следующее:

```
# cat zones/kiev.ua/muff.kiev.ua

$TTL 86400
@    IN     SOA    ns.muff.kiev.ua. admin.muff.kiev.ua. (
                                    2009082400 ; Serial
                                    14400 ; Refresh
                                    7200 ; Retry
                                    3600000 ; Expire
                                    86400 ) ; Minimum



@           IN      NS        ns.muff.kiev.ua.
@           IN      NS        ns2.muff.kiev.ua.

@           IN      MX  10  mail.muff.kiev.ua.
@           IN      MX  20  relay2.muff.kiev.ua.



@           IN      A         195.3.159.250
www         IN      A         195.3.159.250
mail         IN      A         193.227.206.56

relay2      IN      A         193.227.206.57
ns          IN      A         195.3.159.250

ns2         IN      A        193.227.206.50
```

Добавляем несколько строк для старта **BIND** в **/etc/rc.conf**:

```
# echo '### Domain Name Service' >> /etc/rc.conf
 # echo 'named_enable="YES"' >> /etc/rc.conf
 # echo 'named_flags="-u bind -c /etc/namedb/named.conf"' >> /etc/rc.conf
```

Поскольку **BIND** будет работать от имени пользователя **bind**, сделаем его владельцем каталогов и файлов:

```
# chown -R bind:wheel /var/named/log
```

Запускаем демон сервера доменных имен:

```
# sh /etc/rc.d/named start
```

Проверяем, запустился ли named:

```
# ps -ax | grep bind

39512 ?? Is 0:00,05 /usr/sbin/named -u bind -c /etc/namedb/named.conf -t /var/named -u bind
```

Все в норме, **BIND** запустился. Теперь проверим, как он будет отрабатывать наши запросы:

```
# dig A @127.0.0.1 mail.ru
; <<>> DiG 9.4.3-P3 <<>> A @127.0.0.1 mail.ru
; (1 server found)
;; global options: printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 27003
;; flags: qr rd ra; QUERY: 1, ANSWER: 4, AUTHORITY: 6, ADDITIONAL: 0

;; QUESTION SECTION:
;mail.ru. IN A

;; ANSWER SECTION:
mail.ru. 60 IN A 217.69.128.43
mail.ru. 60 IN A 217.69.128.44
mail.ru. 60 IN A 217.69.128.41
mail.ru. 60 IN A 217.69.128.42

;; AUTHORITY SECTION:
mail.ru. 3600 IN NS ns.mail.ru.
mail.ru. 3600 IN NS ns1.mail.ru.
mail.ru. 3600 IN NS ns2.mail.ru.
mail.ru. 3600 IN NS ns4.mail.ru.
mail.ru. 3600 IN NS ns5.mail.ru.
mail.ru. 3600 IN NS ns3.mail.ru.

;; Query time: 162 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Mon Aug 24 02:02:59 2009
;; MSG SIZE rcvd: 196

# dig A @127.0.0.1 muff.kiev.ua

; <<>> DiG 9.4.3-P3 <<>> A @127.0.0.1 muff.kiev.ua
; (1 server found)
;; global options: printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 35632
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 2

;; QUESTION SECTION:
;muff.kiev.ua. IN A

;; ANSWER SECTION:
muff.kiev.ua. 86400 IN A 195.3.159.250

;; AUTHORITY SECTION:
muff.kiev.ua. 86400 IN NS ns.muff.kiev.ua.
muff.kiev.ua. 86400 IN NS ns2.muff.kiev.ua.

;; ADDITIONAL SECTION:
ns.muff.kiev.ua. 86400 IN A 195.3.159.250
ns2.muff.kiev.ua. 86400 IN A 193.227.206.50

;; Query time: 7 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Mon Aug 24 02:02:49 2009
;; MSG SIZE rcvd: 113
```

Демон полноценно функционирует. Отдал запрошенные данные домена **mail.ru** и отдал запрошенные данные для поддерживаемого домена **muff.kiev.ua**.

Проверям работу утилиты **rndc** (чтобы просмотреть допустимые ключи и опции - необходимо набрать команду без ключей)

```
# rndc status

number of zones: 15
debug level: 0
xfers running: 0
xfers deferred: 0
soa queries in progress: 0
query logging is ON
recursive clients: 0/0/1000
tcp clients: 0/100
server is up and running


# rndc reload
server reload successful
```

Теперь дело за малым - указать серверу, чтобы он свои dns-запросы обрабатывал самостоятельно. Изменяем опцию nameserver в файле **/etc/resolv.conf** на адрес loopback-интерфеса сервера:

```
# cat /etc/resolv.conf

domain muff.kiev.ua
nameserver 127.0.0.1
```

Поздравляю. Теперь у вас есть полноценный DNS-сервер на платформе BIND9.

Примечание

Любой файл зоны должен оканчиваться пустой строкой ! Иначе в логах вы будете видеть подобное:

```
# tail -f /var/named/log/general

24-Aug-2009 02:08:17.125 general: warning: /zones/kiev.ua/muff.kiev.ua:25: file does not end with newline
```

Источник (получено *2026-11-07 00:01*):
http://muff.kiev.ua/content/bind-9-nastroika-dns-servera