Konfiguracja serwera DNS w systemie operacyjnym linux

Autorzy: Monika Nazarko, Krzysztof Raczkowski IVFDS

STRESZCZENIE

Celem niniejszego projektu jest przedstawienie możliwości systemu linux w zarządzaniu własną domeną internetową. Opisaliśmy instalację i podstawową konfigurację serwera DNS - programu named z pakietu Bind9. Oprócz opisania budowy plików konfiguracyjnych zamieściliśmy przykładowe ustawienia serwera istniejącego w rzeczywistości - służącego jako serwer DNS dla małej, blokowej sieci lokalnej, posiadającej własną domenę internetową.

SPIS TREŚCI

1. WSTĘP

Zanim zaczniemy konfigurować serwer DNS, należy wytłumaczyć sobie, czym on w ogóle jest i do czego go potrzebujemy.

1.1. Krótko o adresacji komputerów w sieci Internet

Sieć Internet (oraz większość innych sieci) działa w oparciu o protokół IP. Obecnie używana jest jego czwarta wersja (IPv4), choć trwają już prace nad wdrożeniem nowocześniejszej (IPv6). Każdy host (lub urządzenie sieciowe) widziane w sieci posiada swój adres logiczny, dzięki któremu jest identyfikowany. Adresy IP w wersji czwartej mają długość 32 bitów (4 ośmiobitowe oktety) postaci: aaa.bbb.ccc.ddd. Tak więc, korzystając z adresu IP, można znaleźć w sieci wybrany komputer/serwer i skorzystać z udostępnianych przez niego usług.

Jednak ten sposób adresacji nie jest zbyt wygodny dla użytkowników korzystających z sieci komputerowych. Zdolność człowieka do zapamiętywania ciągów cyfr jest znacznie niższa od umiejętności przyswajania wyrazów. Dlatego wprowadzono bardziej przyjazny sposób identyfikacji hostów – **nazwę domenową**.

1.2. Nazwa domenowa

Nazwa domenowa (z ang. *domain name*) składa się z podzielonych kropkami etykiet, określających **domenę internetową**. Etykiety te występują w określonym porządku – od najbardziej szczegółowych, do najbardziej ogólnych.

I tak na przykład, w nazwie news.prz.rzeszow.pl:

- news – oznacza serwer nntp (news),
- prz – Politechnika Rzeszowska,
- rzeszow – miasto,
- pl – kraj (Polska).

1.3. Co to jest domena internetowa?

Aby nie było bałaganu związanego z nazewnictwem hostów w sieci, wprowadzono pewien porządek w jego określaniu. Utworzono pewne hierarchie, według których ustala się nazwę domenową. Ustalono, że podział będzie następował pod względem instytucjonalnym, jak i ze względu na położenie geograficzne.

Powstały więc domeny organizacyjne najwyższego rzędu (TLD – ang. *Top Level Domains*): *com* (dla organizacji handlowych), *edu* (uniwersytety i inne organizacje edukacyjne), *gov* (przedstawicielstwa rządowe), *mil* (organizacje militarne), *net* (ważniejsze centra wspomagające działanie sieci), *int* (organizacje międzynarodowe), *org* (inne organizacje). W ostatnim czasie dodano do powszechnego użytku kilka nowych domen, np.: *name*, *info*, *biz*.

Także poszczególne kraje mają własne domeny: np. *pl* – Polska, *us* – Stany Zjednoczone, *fr* – Francja, *de* – Niemcy.

Wszystkie nazwy nadawane hostom znajdują się wewnątrz innych domen, tworząc tzw. **poddomeny**.

1.4. Poddomena

Poddomena (ang. *subdomain*) to domena znajdująca się poniżej innej domeny. Przykładem może tu być domena *prz.rzeszow.pl*, która jest poddomeną domeny *rzeszow.pl*. Z kolei *rzeszow.pl* to poddomena domeny *pl*.

1.5. A czym jest serwer DNS?

Po wyjaśnieniu podstawowych kwestii, pozostaje pytanie – czym jest serwer DNS? I do czego go potrzebujemy?

Otóż jest to serwer, który zajmuje się dwustronną konwersją numerycznych adresów internetowych (adresów IP) na łańcuchy nazw domenowych. Ich użycie jest konieczne z prostego względu – urządzenia sieciowe komunikują się ze sobą korzystając wyłącznie z adresów w postaci numerycznej. Przyjazne nazwy zostały wprowadzone wyłącznie dla wygody użytkowników.

Każdy serwer posiada dane o adresach komputerów znajdujących się w domenie, którą obsługuje.

1.5.1. Korzenie, czyli root-serwery

Jak już wspomnieliśmy wcześniej, struktura DNS jest hierarchiczna. Na jej szczycie stoi 13 serwerów głównych – tzw. root-serwerów. Przechowują one identyczne dane – adresy IP serwerów będących TLD (*Top Level Domains*) – zarówno organizacyjnych, jak i narodowych. Ponieważ root-serwery działają niezależnie od siebie, minimalizuje to ryzyko awarii całego systemu DNS.

1.5.2. Ale jak to wszystko działa?

Przykładowo, chcemy się połączyć ze stroną internetową www.prz.rzeszow.pl. Aby nawiązać połączenie, nasz system musi poznać adres IP docelowego serwera. Wysyła więc zapytanie do obsługującego go serwera DNS (zwykle serwer udostępniany przez providera). Ten (o ile nie posiada wyniku naszego zapytania w swoim cache'u) zwraca się do jednego z root-serwerów z zapytaniem o adres serwera (lub listę adresów serwerów) utrzymującego domenę *pl*. W następnej kolejności odpytywany jest serwer domeny *pl*, który zwraca adres serwera odpowiedzialnego za domenę *rzeszow.pl*. Ten z kolei kieruje do serwera przechowującego dane domeny *prz.rzeszow.pl*, a z tego ostatniego serwera otrzymujemy adres komputera o szukanej przez nasz nazwie.

2. INSTALACJA SERWERA DNS W SYSTEMIE OPERACYJNYM LINUX – PAKIET BIND9

Najpopularniejszym serwerem DNS w systemie linux jest *named*, znajdujący się w pakiecie Bind. Obecną wersją stabilną pakietu jest wersja 9. W naszym opracowaniu korzystaliśmy z wydania 9.2.1.

2.1. Instalacja gotowego pakietu z binariami

2.1.1. Debian Wody (pakiety *deb*)

W poniższym opracowaniu wykorzystywaliśmy dystrybucję linuksa o nazwie Debian w wersji 3.0 (Woody). Korzysta on z systemu gotowych pakietów (pliki *deb*), powiązanych ze sobą szeregiem zależności. Całością zarządza program *apt-get*. Przy jego pomocy można zainstalować pakiet bind9:

```
julia:~# apt-get install bind9
```

Po zainstalowaniu, named jest automatycznie uruchamiany, wykorzystując domyślny plik konfiguracyjny (pozwalający wykorzystywać serwer DNS jako serwer cache'ujący). Konfigurację można dopasować do własnych potrzeb.

2.1.2. Red Hat, Mandrake lub PLD (pakiety *rpm*)

Dystrybucje takie jak Red Hat, Mandrake czy PLD wykorzystują pakiety *rpm*. Aby zainstalować w nich pakiet bind, należy go ściągnąć i wywołać komendę:

```
tux:~# rpm -ivh bind-9.2.1*
```

2.2. Instalacja ze źródeł

Aby zainstalować pakiet bind korzystając z jego źródeł, należy najpierw je zdobyć (najnowsze dostępne są pod adresem ftp://ftp.isc.org/isc/bind9/9.2.1/bind-9.2.1.tar.gz). Można też skorzystać z pakietu *deb* lub *rpm* ze źródłami.

Po rozpakowaniu źródeł (komendą *tar -xvzf nazwa_archiwum*) lub zainstalowaniu odpowiedniego z nimi pakietu, należy przejść do utworzonego katalogu (*cd ścieżka/do/katalogu/ze/źródłami*) i zapoznać się z plikiem *readme*, wyjaśniającym szczegóły kompilacji.

Standardowo, kompilacja przebiega według następujących kroków:

- Uruchamiamy skrypt konfiguracyjny: *./configure* (możemy dodać odpowiednie opcje konfiguracyjne – szczegóły *./configure --help*)
- Kompilujemy źródła: *make*
- Instalujemy skompilowane pliki binarne: *make install*.

3. PODSTAWOWA KONFIGURACJA SERWERA NAZW – PLIK NAMED.CONF

Plikiem konfiguracyjnym serwera DNS *named* jest plik *named.conf*. Jego domyślna lokalizacja to zwykle */etc/named.conf* (ale np. Debian standardowo kopiuje ten plik do */etc/bind/named.conf*). Definiuje on działanie serwera. *Named* korzysta także z plików stref, w których zapisane są dane o obsługiwanych przez serwer domenach (a także adresy root-serwerów). Szczegółowe omówienie plików stref znajduje się w punkcie 4 tego opracowania.

3.1. Budowa pliku named.conf

Plik ten składa się z bloków wyrażeń. Każdy blok zbudowany jest następująco:

```
wyrażenie [nazwa] {
opcje_dla danego_wyrażenia;
...
};
```

Podstawowe wyrażenia, niezbędne do prawidłowego skonfigurowania serwera, opiszemy w kolejnych punktach.

Każdy blok oraz każda opcja kończą się średnikiem.

Dopuszczalne jest stosowanie komentarzy – tekstów ignorowanych przez serwer w czasie odczytu konfiguracji. Muszą one zostać oznaczone. Dopuszcza się trzy rodzaje oznaczeń dla komentarzy:

```
# jak w shellu
```

```
// jak w C/C++
/* jak w C/C++ */
```

Nie można oznaczać komentarzy za pomocą średnika *;* - jest on zarezerwowany jako znak końca wyrażenia.

3.1.1. Wyrażenie acl

Wyrażenie to tworzy listy kontroli dostępu (ang. *access control list*), pozwalające na zdefiniowanie adresów hostów, którym zezwalamy (bądź zabraniamy) na korzystanie z serwera lub części udostępnianych przez niego danych.

Wyrażenie to ma ogólną postać:

```
acl nazwa_ograniczenia {
adres[/maska];
adres[/maska];
...
}
```

Każdy wpis może być poprzedzony znakiem [*!*] (wykrzyknik), oznaczającym zaprzeczenie.

Podanie maski sieci jest niezbędne, jeżeli ograniczenie ma dotyczyć całej zdefiniowanej sieci (np. sieć 192.168.0.0/24). Jest opcjonalne, jeżeli wprowadzony do listy adres dotyczy pojedynczego hosta (np. wpis 192.168.0.12/32 jest równoznaczny wpisowi 192.168.0.12).

Składnikiem listy może być inna, zdefiniowana wcześniej lista.

Istnieją cztery predefiniowane listy dostępowe:

- any – dowolny adres,
- none – żaden adres,
- localhost – adres interfejsu komputera, na którym działa *named*,
- localnet – dowolny adres z sieci należącej do tej, w której działa *named*.

Przykładowa lista dostępowa może mieć postać:

```
acl dozwolone {
!172.16.31.1;
172.16.31.0/24
};
```

co oznacza zezwolenie wszystkim hostom z sieci 172.16.31.0/24 oprócz hosta 172.16.31.1.

albo (z wykorzystaniem poprzednio zdefiniowanej listy):

```
acl dozwolone2 {
dozwolone;
192.168.10.2;
```

192.168.10.4;

};

gdzie do poprzedniej listy dołączono zezwolenie dla dwóch nowych hostów.

3.1.2. Wyrażenie options

Wyrażenie to pozwala na zdefiniowanie globalnych opcji konfiguracyjnych serwera. Jego ogólna postać to:

options {

opcja wartość;

...

};

Do najważniejszych opcji należą:

- directory „ścieżka/do/katalogu/z/plikami/stref”;

Określa ścieżkę do katalaogu, w którym przechowywane są pliki stref, np.:

directory "/var/cache/bind";

- query-source address <adres_IP> port <nr_portu>;

Określa numer portu, z którego będą wychodzić zapytania kierowane do innych serwerów DNS (standardowo używany jest jeden z nieuprzywilejowanych portów – powyżej 1023). Przydatne, jeśli serwer chroniony jest zaporą ogniową, np.:

query-source address * port 53;

- listen-on <port nr_portu> { lista_adresów };

Pozwala na uruchomienie kilku instancji serwera, z których każda może nasłuchiwać na innym porcie zapytań z interfejsów określonych w liście dostępu, np. na standardowym porcie 53 nasłuch od hosta 172.16.31.4, na porcie 1234 dla sieci 192.168.0.0/16 (z wyjątkiem adresu 192.168.1.7):

listen-on { 172.16.31.4; };

listen-on port 1234 { !192.168.1.7; 192.168/16; };

- allow-query { lista_dostępowa };

Określa adresy, spod których przyjmowane są zapytania do serwera. Opcja ta wykorzystuje zdefiniowane wcześniej listy dostepowe (*access list*), np. zezwolenie na korzystanie z zasobów serwera tylko sieci lokalnej:

allow-query { localnet; };

lub adresom z uprzednio zdefiniowanej listy dozwolone2:

allow-query { "dozwolone2”; };

- recursion [yes | no];

Określa, czy serwer może kontaktować się z innymi serwerami w celu znalezienia wyniku

zapytania klienta. Domyślna wartość to *yes*.

Więcej użytecznych opcji znaleźć można w „BIND 9 Administrator Reference Manual”.

3.1.3. Wyrażenie zone

Wyrażenie to służy do definiowania stref obsługiwanych przez serwer. Jego składnia wygląda następująco:

```
zone "nazwa.strefy” {
  type typ_strefy;
  <opcje>;
  …
};
```

Typem strefy może być:

- hint – określa listę root-serwerów,
- master – stwierdza, że serwer jest głównym (podstawowym) serwerem dla danej strefy,
- slave – serwer jest serwerem pomocniczym (zapasowym) dla danej strefy – uaktualnia zawartość strefy korzystając z danych serwera głównego,
- forward – serwer jest serwerem przekazującym zapytania dla podanej strefy do innych, zdefiniowanych parametrem *forwarders* serwerów,
- stub – podobny do *slave*, jednakże serwer uaktualnia tylko rekordy NS.

Najważniejsze z pozostałych opcji to:

- file "nazwa.pliku.strefy";

Określa nazwę pliku przechowującego informacje o strefie.

- allow-query { lista_dostępowa };

Działa tak samo jak parametr z sekcji *options*.

- masters <port nr_portu> { lista_serwerów };

Wyłącznie w strefie typu *slave* – ustala adresy serwerów głównych dla danej strefy, z którymi łączy się serwer w celu pobrania uaktualnienia strefy.

- notify [yes | no];

Określa, czy serwer podstawowy ma informować serwery pomocnicze o uaktualnieniu strefy.

- also-notify { lista_adresów };

Za pomocą tej opcji serwer będzie powiadamiał inne serwery (z listy adresów, a niebędące wymienionymi w rekordzie NS strefy) o zmianach w strefie.

- allow-transfer { lista_adresów };

To lista adresów serwerów, które mogą zażądać transferu całej strefy. Muszą tu być wpisane

serwery będące pomocniczymi dla obsługiwanej przez serwer strefy.

- allow-update { lista_adresów };

Lista serwerów, które mogą aktualizować obsługiwana strefę.

Przykładem zdefiniowanej sekcji *zone* może być poniższy wpis:

```
zone "b4net.int.pl"{
        type master;
        file "b4net.int.pl";
        allow-update { none; };
        allow-transfer { 62.233.151.50; };
        notify yes;
};
```

Strefa ma nazwę *b4net.int.pl*, serwer jest jego głównym serwerem (master), plik strefy ma nazwę *b4net.int.pl* (znajduje się w katalogu zdefiniowanym w sekcji *options/directory*). Żaden serwer nie może uaktualniać strefy (może to zrobić tylko administrator zarządzający danym serwerem), a jej zawartość może być pobrana przez serwer o adresie *62.233.151.50*. Serwer zapasowy jest powiadamiany o zmianie strefy.

3.1.4. Odwrotny DNS

Odwrotny DNS (reverse DNS) służy do powiązania adresu IP z jego nazwą domenową (czyli odwrotnie niż w przypadku zwykłego zapytania). Do konfiguracji rev DNS służy strefa odwrotna, np.:

```
zone "0.0.127.in-addr.arpa" {
  type master;
  file "named.local";
};
```

Jest to strefa odpowiadająca na zapytanie o nazwę dla adresów z sieci 127.0.0.x. Jak widać, nazwa strefy zaczyna się od przestawienia wspak adresu, któremu odpowiada strefa oraz dodaniu słów „in-addr.arpa”. Pozostała część konfiguracji strefy jest identyczna jak w przypadku zwykłej strefy.

3.2. Serwer cache’ujący (caching-only server)

Często zdarza się, że serwer nie obsługuje żadnej strefy, a służy wyłącznie do rozwiązywania nazw domenowych będących zapytaniami klientów (np. serwer ISP udostępniającego łącze internetowe sieci lokalnej). Serwer taki ma za zadanie odciążanie ruchu wychodzącego z sieci LAN, poprzez buforowanie wyników zapytań w celu ponownego udostępniania ich klientom.

W przypadku, gdy odpowiedź na zapytanie klienta znajduje się już w cache’u serwera, nie ma potrzeby przekazywania zapytania dalej (nie wychodzi ono poza obręb sieci lokalnej). Gdy cache nie zawiera potrzebnych danych, serwer wysyła zapytanie do innych serwerów, korzystając z adresów serwerów głównych (root-serwerów), o których informacje zapisane są w strefie

typu *hint*. Odpowiedź zapisywana jest w lokalnym buforze i może być wykorzystywana przez inne pytające komputery.

Jeśli tak działający serwer DNS znajduje się w sieci lokalnej z dostępem do Internetu, odciąża on ruch na wąskim gardle, jakim jest łącze sieci LAN ze światem.

Aby serwer działał jako serwer cache'ujący, musi mieć odpowiednio skonfigurowaną strefę serwerów głównych ".":

```
zone "." {
    type hint;
    file "/etc/bind/db.root";
};
```

Plik *db.root* powinien ponadto zawierać listę dostępnych root-serwerów. Jak powinien wyglądać plik strefy serwerów głównych, przedstawia punkt 4.2.

3.3. Serwer autorytatywny

Może zajść potrzeba, aby serwer odpowiadał na zapytania dotyczące wyłącznie stref, które obsługuje. Można w tym celu wyłączyć obsługę zapytań kierowanych do innych serwerów dla żądań nie znalezionych w lokalnym buforze serwera. Służy do tego parametr *recursion* w sekcji *options*. Przykładowa konfiguracja może wyglądać tak:

```
options {
        directory "/var/named";
        recursion no;
};
zone "b4net.int.pl"{
        type master;
        file "b4net.int.pl";
        allow-update { none; };
        allow-transfer { 62.233.151.50; };
        notify yes;
};
```

Serwer skonfigurowany jest do przyjmowania zapytań wyłącznie o strefę *b4net.int.pl*.

3.4. Serwer forwardujący zapytania

Ten typ serwera służy jako pośrednik pomiędzy pytającymi klientami a wybranymi serwerami, do których przekazywane są wszelkie zapytania nie znalezione w lokalnym buforze. Pozwala to wybrać inne serwery niż główne (root) do zapytań o niezbuforowane domeny. Rozwiązanie przyspieszy działanie serwera, w przypadku gdy występują duże opóźnienia przy połączeniu z root-serwerami, a jest możliwość skorzystania z zasobów cache'a serwera z bliższej lokacji (zwykle o szybszym czasie odpowiedzi).

Aby skonfigurować serwer do forwardowania zapytań, wystarczy odpowiednio ustawić stre-

fę ".":

```
zone "." IN {
        type forward;
        forward only;
        forwarders { 194.204.159.1; 194.204.152.34; };
};
```

Powyższy przykład wymusza (*forward only*) zamiast korzystania z serwerów głównych, przesyłanie zapytań do serwerów nazw TPSA (dns.tpsa.pl i dns2.tpsa.pl).

3.5. Serwer podstawowy i zapasowy (master/slave)

Każda zarejestrowana domena powinna posiadać co najmniej dwa serwery przechowujące dane o jej strefie. Do prawidłowego działania domeny wystarczy co prawda jedynie jeden serwer, ale między innymi ze względów bezpieczeństwa (redundacja na wypadek awarii) wskazane jest posiadanie serwera (kilku serwerów) zastępczego.

Po każdej modyfikacji pliku strefy, serwer podstawowy powiadamia serwery zapasowe (zdefiniowane w rekordzie NS strefy oraz przy pomocy opcji *also-notify*) o konieczności aktualizacji wpisu. Wówczas serwery slave proszą o transfer strefy w celu uaktualnienia.

Serwer będący podstawowym dla jednej strefy może być zarazem zapasowym dla innej. Definiowanie serwera mastera i slave'a może wyglądać następująco:

```
zone "amigos.kom.pl" {
   type master;
   file "amigos.kom.pl";
   allow-update { none; };
   allow-transfer { 62.233.151.50; };
   notify yes;
      };
```

Jest to serwer podstawowy (*master*) dla domeny amigos.kom.pl, jego serwerem zapasowym jest komputer o adresie *62.233.151.50*.

```
zone "b4net.int.pl" {
      type slave;
      file "/var/named/b4net.int.pl";
      masters { 217.97.21.84; };
};
```

A to jest serwer zapasowy (*slave*) dla domeny b4net.int.pl. Jego serwerem podstawowym jest komputer o adresie *217.97.21.84*.

4. PLIKI STREF

Pliki stref przechowują dane opisujące obsługiwane przez serwer strefy. Są one wczytywane w czasie uruchamiania serwera i następnie używane do udzielania odpowiedzi na kierowane

zapytania.

Dane w plikach stref zapisywane są w rekordach. Różne rekordy służą do definiowania różnych typów danych.

4.1. Budowa plików stref

Wpisy w pliku stref mają ogólna postać:

<nazwa> IN [<TTL>] typ_rekordu wartość

<nazwa> odnosi się do nazwy domeny, nazwy komputera lub adresu IP (dla strefy odwrotnej).

<TTL> to czas życia rekordu – okres, przez jaki ma być buforowany wpis o danym rekordzie (opcjonalne).

Typ rekordu określa rodzaj zdefiniowanego rekordu (alias, serwer DNS, inne).

Wartość to zwykle adres IP, jaki chcemy skojarzyć z określoną nazwą.

Jeśli zamiast nazwy (bądź wartości pola) podamy znak @, w jego miejsce zostanie wstawiona nazwa pliku strefy lub wartość dyrektywy $ORIGIN (ustawianej zwykle na początku pliku strefy).

Do nazwy nie kończącej się kropką dopisywana jest domyślna nazwa domenowa zdefiniowana w nazwie pliku strefy (bądź dyrektywie $ORIGIN), np. dla pliku strefy „*b4net.int.pl*” wpis:

www IN A 217.97.21.84

jest równoznaczny z wpisem:

www.b4net.int.pl. IN A 217.97.21.84

W pliku stref można korzystać z komentarzy, które oznaczane są poprzez znak #:

to jest komentarz

4.1.1. Dyrektywy

Dyrektywy znajdują się na początku pliku strefy. Są to komendy ułatwiające prace z plikiem strefy.

Do najważniejszych wykorzystywanych dyrektyw należą:

- $TTL

Pozwala na określenie domyślnego czasu życia rekordów strefy. Parametr ten informuje bufor serwera DNS, jak długo ma trzymać dane o pobranym rekordzie.

- $ORIGIN

Określa nazwę domyślnej domeny. Jeżeli wpis nazwy w rekordzie nie kończy się kropką, wartość tej dyrektywy dołączana jest do jej końca. Również znak @ zastępuje domyślną nazwę określoną za pomocą tej dyrektywy.

- $INCLUDE

Pozwala na dołączenie do pliku strefy inny plik (np. z dalszymi danymi o strefie).

4.1.2. Rekord SOA

Rekord SOA (*Start Of a zone of Authority*) rozpoczyna każdy plik strefy. Jego składnia wy-

gląda następująco:

<domena> [<TTL>] IN SOA <MNAME> <RPERSON> (
<SERIAL>
<REFRESH>
<RETRY>
<EXPIRE>
<MINIMUM>
)

<MNAME> to główny serwer nazw dla konfigurowanej domeny (nie jest do niego przesyłane powiadomienie o aktualizacji strefy, nawet gdy jest on wpisany w rekordzie NS domeny).

<RPERSON> to adres e-mail osoby odpowiedzialnej za domenę (zamiast znaku @ występującego w adresie e-mail używamy tu kropki).

<SERIAL> to numer seryjny rekordu. Zwykle w postaci yyyymmddnn, gdzie pierwsza część to data aktualizacji, zaś nn to numer modyfikacji wykonany tego dnia.

<REFRESH> to czas, co jaki pomocnicze serwery DNS maja odświeżyć informacje o strefie.

<RETRY> to czas, po którym zapasowy serwer ma podjąć ponowną próbę odświeżenia informacji o domenie w przypadku niepowodzenia.

<EXPIRE> to czas, po którym zapasowy serwer powinien przestać odpowiadać na zapytania kierowane do domeny, jeżeli nie można się skontaktować z jej serwerem podstawowym.

<MINIMUM> to czas przechowywania przesłanych danych o nieistniejącej domenie na innych serwerach DNS.

4.1.3. Jednostki czasu

Standardową jednostką czasu określaną w dyrektywie $TTL i rekordzie SOA jest sekunda. Istnieje jednak możliwość wyrażenia czasu w innych jednostkach:

- M – minuta,
- H – godzina,
- D – dzień,
- W – tydzień.

4.1.4. Rekordy MX

Rekord ten służy do wskazania serwerów pocztowych obsługujących konfigurowaną domenę. Składa się z dwóch pól – priorytetu serwera i jego nazwy:

IN MX <priorytet> <nazwa>

Priorytet określa pierwszeństwo w wyborze serwera poczty (w przypadku gdy jest ich zdefiniowanych kilka). Dwa serwery o tym samym priorytecie będą wykorzystywane w sposób losowy.

Przykład rekordu MX:

IN MX 10 mail.b4net.int.pl.

Każdy rekord MX powinien posiadać odpowiedni rekord typu A, wskazujący na adres IP

zdefiniowanego serwera. Dla naszego przykładu rekord ten może mieć postać:

mail.b4net.int.pl IN A 217.97.21.84

4.1.5. Inne typy rekordów

Najczęściej wykorzystywane typy rekordów to:

- A – określa adres IP dla podanej nazwy,
- CNAME – nazwa kanoniczna dla aliasu,
- NS – serwer DNS przechowujący dane o domenie,
- PTR – w strefach odwrotnych określa nazwę komputera (przypisaną do adresu IP).

4.2. Plik strefy serwerów głównych

Jest to plik przechowujący dane dla strefy ".". Zawiera informacje o adresach wszystkich trzynastu głównych serwerów nazw. Plik ten wygląda następująco:

```
\&.             6D IN NS    G.ROOT-SERVERS.NET.
\&.             6D IN NS    J.ROOT-SERVERS.NET.
\&.             6D IN NS    K.ROOT-SERVERS.NET.
\&.             6D IN NS    L.ROOT-SERVERS.NET.
\&.             6D IN NS    M.ROOT-SERVERS.NET.
\&.             6D IN NS    A.ROOT-SERVERS.NET.
\&.             6D IN NS    H.ROOT-SERVERS.NET.
\&.             6D IN NS    B.ROOT-SERVERS.NET.
\&.             6D IN NS    C.ROOT-SERVERS.NET.
\&.             6D IN NS    D.ROOT-SERVERS.NET.
\&.             6D IN NS    E.ROOT-SERVERS.NET.
\&.             6D IN NS    I.ROOT-SERVERS.NET.
\&.             6D IN NS    F.ROOT-SERVERS.NET.

G.ROOT-SERVERS.NET.     5w6d16h IN A    192.112.36.4
J.ROOT-SERVERS.NET.     5w6d16h IN A    198.41.0.10
K.ROOT-SERVERS.NET.     5w6d16h IN A    193.0.14.129
L.ROOT-SERVERS.NET.     5w6d16h IN A    198.32.64.12
M.ROOT-SERVERS.NET.     5w6d16h IN A    202.12.27.33
A.ROOT-SERVERS.NET.     5w6d16h IN A    198.41.0.4
H.ROOT-SERVERS.NET.     5w6d16h IN A    128.63.2.53
B.ROOT-SERVERS.NET.     5w6d16h IN A    128.9.0.107
C.ROOT-SERVERS.NET.     5w6d16h IN A    192.33.4.12
D.ROOT-SERVERS.NET.     5w6d16h IN A    128.8.10.90
```

```
E.ROOT-SERVERS.NET.     5w6d16h IN A    192.203.230.10
I.ROOT-SERVERS.NET.     5w6d16h IN A    192.36.148.17
F.ROOT-SERVERS.NET.     5w6d16h IN A    192.5.5.241
```

4.3. Plik strefy dla utrzymywanej domeny

Przykładowy plik strefy dla domeny:

```
$TTL 86400
$ORIGIN amigos.kom.pl.

@       IN      SOA     amigos.kom.pl. root.amigos.kom.pl. (
                        2002072605
                        21600
                        7200
                        1209600
                        86400 )

@       IN      NS      pj85.krakow.sdi.tpnet.pl.
@       IN      NS      naskarpie.futuro.pl.
dns     IN      A       217.97.124.85

@       IN      MX      10      mail
mail    IN      A       217.97.124.85

@       IN      A       217.97.124.85

www     IN      A       217.97.124.85

ftp     IN      A       217.97.124.85
```

5. PRZYKŁADOWA KONFIGURACJA SERWERA DNS

Poniżej zamieszczamy listing plików konfiguracyjnych działającego serwera DNS, utrzymującego domenę b4net.int.pl:

Plik /etc/bind/named.conf

```
options {
        directory "/etc/bind";
        auth-nxdomain yes;
```

```
};

zone "." IN {
        type forward;
        forward only;
        forwarders { 194.204.159.1; 194.204.152.34; };
};

zone "0.0.127.in-addr.arpa" {
        type master;
        file "named.local";
};

zone "b4net.int.pl"{
        type master;
        file "b4net.int.pl";
        allow-update { none; };
        allow-transfer { 62.233.151.50; };
        notify yes;
};
```

Plik /etc/bind/named.local:

```
$TTL    86400
@       IN      SOA     localhost. root.localhost.  (
                                1997022700 ; Serial
                                28800      ; Refresh
                                14400      ; Retry
                                3600000    ; Expire
                                86400 )    ; Minimum
            IN      NS      localhost.

1       IN      PTR     localhost.
```

Plik /etc/bind/b4net.int.pl:

```
$TTL 86400
$ORIGIN b4net.int.pl.
```

```
@       IN      SOA     b4net.int.pl. root.b4net.int.pl. (
        2002072605
        21600
        7200
        1209600
        86400 )

@       IN      NS      pf84.rzeszow.sdi.tpnet.pl.
@       IN      NS      naskarpie.futuro.pl.
dns     IN      A       217.97.21.84

@       IN      MX      10      mail
mail    IN      A       217.97.21.84

@       IN      A       217.97.21.84

www     IN      A       217.97.21.84

ftp     IN      A       217.97.21.84

private IN      A       217.97.21.84

conference IN A         217.97.21.84

jud     IN      A       217.97.21.84

gg      IN      A       217.97.21.84
```

6. PODSUMOWANIE

Opracowanie to pozwala w podstawowy sposób zapoznać się ze sposobem konfigurowania serwera DNS w systemie linux. Nie wyczerpuje wszystkich dostępnych opcji, zapewnia jednak prawidłowe działanie serwera i wykorzystanie wszystkich niezbędnych jego funkcji.

Zainteresowanych dokładniejszym poznaniem tajników Binda zapraszamy do lektury jego dokumentacji.

Literatura

[1] BIND 9 Administrator Reference Manual

[2] DNS How-To http://www.jtz.org.pl/Html/DNS-HOWTO.pl.html

[3] Konfiguracja serwera DNS Bind9 http://bogdan.agh.edu.pl/referaty/dns