Podstawowe mechanizmy zabezpieczeń usługi DNS

Usługa Domain Name Service jest jedną z podstawowych usług sieciowych koniecznych do funkcjonowania sieci opartych na stosie protokołów IP (w tym Internet) w ich obecnym kształcie.
Stanowi ona wysoce skalowalną, rozproszoną bazę danych, pozwalającą na, między innymi:

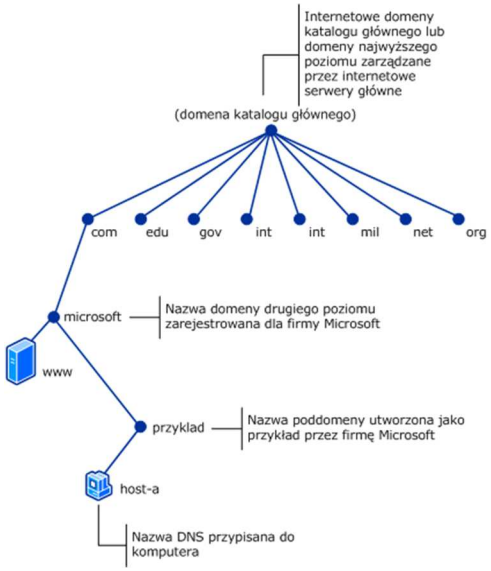
- określanie adresów IP na podstawie nazw DNS,
- określanie nazw DNS odpowiadających określonym adresom IP,
- wyszukiwanie usług, świadczących je serwerów oraz podstawowej informacji konfiguracyjnej koniczej do skorzystania z nich.

Dokładny opis usługi DNS nie jest przedmiotem tego opracowania, lecz poniżej przypomniane zostaną podstawowe pojęcia związane opisywaną usługą.

1. DNS jako rozproszona baza danych

W celu zapewnienia skalowalności oraz możliwości rozproszonego zarządzania, baza danych usługi DNS jest hierarchiczną bazą rozproszoną, zawierającą ponadto elementy redundancji.

Przestrzeń nazw DNS podzielona jest na hierarchicznie ułożone tzw. **domeny** lub **strefy**, obejmujące nazwy kończące się określonym ciągiem znaków, np.: pl, edu.pl, pg.gda.pl, itp.
Hierarchia nazw DNS ma postać drzewa. U jego korzenia leży tzw. domena katalogu głównego (zwana też **domeną root**), zaznaczana znakiem kropki „.”.
Dalej zlokalizowane są tzw. **domeny pierwszego poziomu**, wprowadzające lokalizacyjny lub funkcjonalny podział nazw DNS. Np. pl, cz, uk, … – domeny narodowe (podział lokalizacyjny); edu, com, net (podział funkcjonalny).
Domeny drugiego poziomu są z kolei rejestrowane przez organizacje lub osoby prywatne np.: wp.pl, gdansk.pl, itp.
Organizacja nazw w obrębie domeny zarejestrowanej przez określonego właściciela zależy od jego potrzeb i podjętych decyzji organizacyjnych.



Serwery autorytatywne i transfery stref

Baza danych DNS przechowywana jest na wielu serwerach. Jeśli dany serwer przechowuje bazę danych określonej domeny DNS, to jest w stanie jednoznacznie odpowiedzieć na zapytania jej dotyczące i nazywany jest **serwerem autorytatywnym** dla danej domeny.

Serwer DNS przechowujący bazę danych danej domeny, może być ponadto serwerem **typu master (głównym)** lub typu **slave (podrzędnym)** tej domeny. Kopia bazy danych przechowywana na serwerze głównym, pozwala na wprowadzanie zmian, natomiast serwery podrzędne odpowiadają za utrzymanie swoich kopii bazy danych zsynchronizowanych z serwerem głównym. W tym celu okresowo weryfikują one numer seryjny bazy danych danej strefy na serwerze głównym i w przypadku wykrycia zmiany inicjują tzw. **transfer strefy (zone transfer)**, czyli pobierają nową wersję bazy zastępując własną.

Transfer strefy może mieć charakter całościowy, gdy pobierana jest cała treść bazy – nosi wówczas nazwę **Authoritative Transfer (AXFR)**. Alternatywnie pobierane mogą być wyłącznie zmiany różniące bazy danych o numerach seryjnych serwera podrzędnego i głównego – wówczas mówimy o transferze typu **Incremental Zone Transfer (IXFR)**.

Często spotykaną funkcjonalnością głównych serwerów DNS, jest też wysyłanie powiadomień do serwerów podrzędnych w przypadku gdy baza danych strefy ulegnie modyfikacjom. Pozwala to serwerem podrzędnym zainicjować transfer strefy natychmiast po wprowadzeniu zmian na serwerze głównym.

Delegacja domen

W celu zachowania funkcjonowania bazy danych DNS, baza każdej domeny musi zawierać odwołania wskazujące serwery DNS autorytatywne dla wszystkich jej domen podrzędnych. Na przykład domena pl będzie zawierała odwołania pozwalające określić jakie serwery są autorytatywne dla domen edu.pl, gda.pl, com.pl, itd. Jest to realizowane poprzez umieszczenie w bazie danych strefy odpowiednich rekordów typu NS (patrz 5.1.4) i nosi nazwę delegacji domeny.

Dzięki temu, serwer autorytatywny dla danej domeny zawsze będzie w stanie podać:

- ścisłą i ostateczną odpowiedź na pytania dotyczące domeny w której jest autorytatywny,
- informację jaki serwer jest w stanie podać dokładniejszą informację w przypadku pytań dotyczących domen podrzędnych w stosunku do domeny dla której jest autorytatywny.

Fully Qualified Domain Name (FQDN)

Pełna nazwa DNS, umożliwiająca jednoznaczne zidentyfikowanie poszukiwanej informacji w bazie danych DNS nosi nazwę **Fully Qualified Domain Name (FQDN)** – np. komp1.lab.local. Przeciwieństwem FQDN jest skrócona nazwa DNS, obejmująca jedynie pierwszy jej człon (najniższy w hierarchii DNS i identyfikująca dany rekord w bazie jedynie obrębie przyjętej domeny – np. komp1. W jej przypadku musimy ponadto posiadać informację, w obrębie jakiej domeny DNS należy poszukiwać podanej nazwy. Określenie tzw. domeny domyślnej jest elementem konfiguracji klienta DNS (zwanego inaczej resolverem), który na tej podstawie uzupełnia skrócone nazwy DNS do postaci FQDN przed wysłaniem zapytań do serwera. Klient DNS może wysłać wiele takich zapytań dla jednej nazwy prostej, jeśli skonfigurowano więcej niż jedną domenę domyślną.

Zapytania DNS

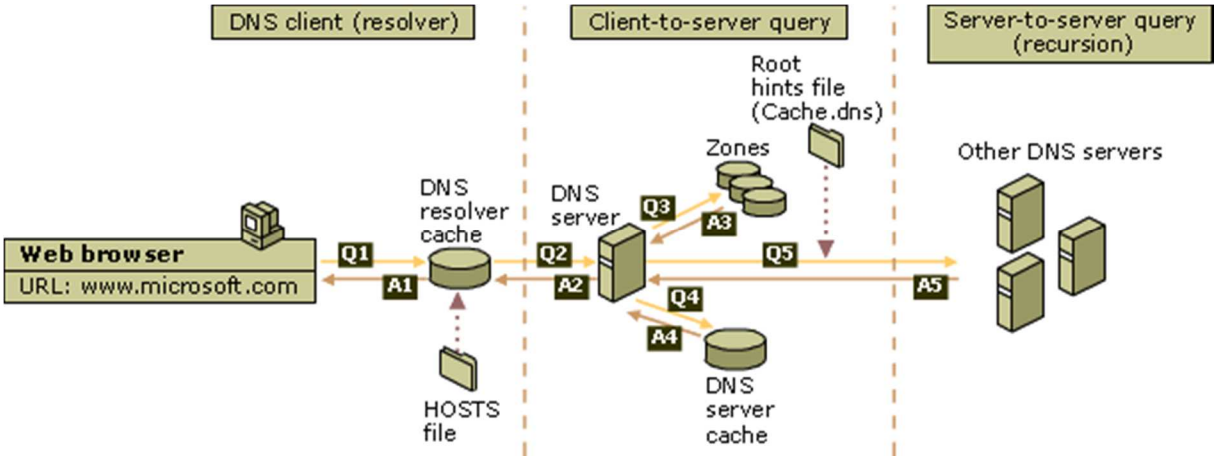
Zapytania DNS dzielimy na dwa podstawowe rodzaje:

- **rekurencyjne** – wysyłający zapytanie jest zainteresowany uzyskaniem ostatecznej odpowiedzi na zadane pytanie – tzn. treścią odpowiedniego rekordu (lub wielu rekordów) z bazy DNS.

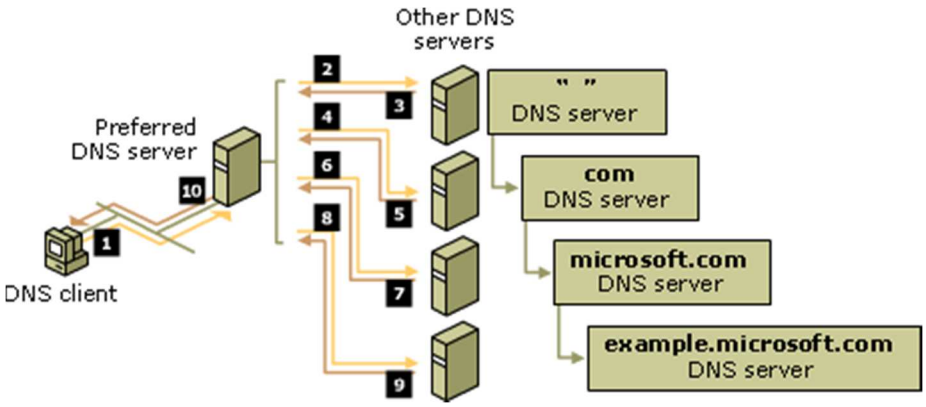
- **iteracyjne** – wysyłający zapytanie akceptuje odpowiedzi „z odwołaniem” (**referral**), które nie zwierają treści rekordu/rekordów o które zapytano, lecz jedynie informacje o innym serwerze DNS, który powinien posiadać dokładniejsze informacje na ten temat od serwera który wysłał taką odpowiedź.

Klienci DNS posługują się najczęściej zapytaniami rekurencyjnymi, które wysyłają do pojedynczego serwera wskazanego w ich konfiguracji.
Serwery DNS posługują się najczęściej zapytaniami iteracyjnymi, próbując uzyskać odpowiedź, której zażądał klient przysyłając im zapytanie rekurencyjne.

Najczęściej proces udzielania odpowiedzi na zapytanie klienta DNS przebiega następująco:



Klient DNS otrzymuje od aplikacji żądanie uzyskania informacji z bazy DNS – np. adresu IP związanego z daną nazwą DNS.
Klient DNS sprawdza czy nie posiada odpowiednej informacji w swojej **pamięci podręcznej** (**DNS resolver cache** – do której przechowywane są przez określony czas informacje uzyskane w wyniku poprzednich zapytań DNS). Jeśli dostępne tam dane pozwalają na udzielenie odpowiedzi, jest ona wysyłana do aplikacji która zainicjowała żądanie.
Jeśli odpowiedzi nie można udzielić na podstawie pamięci podręcznej, do zdefiniowanego w konfiguracji resolvera serwera DNS wysyłane jest zapytanie rekurencyjne (**Q2**).
Serwer otrzymujący takie zapytanie sprawdza najpierw (**Q3**), czy jest autorytatywny dla danej domeny DNS. Jeśli tak, to udziela ostatecznej odpowiedzi bazując wyłącznie na zawartości swojej kopii bazy danej domeny. Jeśli nie, to przeszukuje swoją pamięć podręczną (**Q4**), funkcjonującą na podobnej zasadzie jak pamięć podręczna resolvera DNS i, jeśli jest to możliwe, udziela odpowiedzi na jej podstawie.
Jeśli na żadnym z powyższych etapów nie udało się udzielić odpowiedzi na otrzymane żądanie, serwer rozpoczyna proces wyszukiwania potrzebnych danych w trybie iteracyjnym (**Q5**). Niezbędne w tym procesie adresy serwerów dla domeny root (tzw. **root hints**), serwer uzyskuje z pliku będącego częścią jego konfiguracji.
Na poniższym rysunku zobrazowano powyższy proces dla przykładowego zapytania o nazwę DNS client.example.microsoft.com.



W przypadku wyszukiwania odpowiedzi w trybie iteracyjnym, serwer DNS kieruje zapytanie do jednego serwerów **root** (2). W jego wyniku, uzyskuje on odpowiedź „z odwołaniem" (referral), wskazującą serwer autorytatywny dla domeny pierwszego poziomu, której poddomeną (bezpośrednią lub na którymś z niższych poziomów hierarchii) jest domena zawierająca poszukiwane dane. Informacja ta jest uzyskiwana z rekordów NS (delegacji).
W naszym przykładzie, w odpowiedzi na zapytanie o **client.example.microsoft.com** (2) serwer root zwrócił (3) informacje pozwalające na skierowanie zapytania do serwera autorytatywnego dla domeny **com**.
Serwer kieruje zatem zapytanie o **client.example.microsoft.com** do wskazanego serwera autorytatywnego dla domeny **com** (4). W odpowiedzi (5) uzyskuje referral do serwera autorytatywnego dla domeny **microsoft.com**.
Dalej proces powyższy powtarza się, aż do momentu uzyskania odwołania (referral) do serwera autorytatywnego dla strefy **example.microsoft.com**. Serwer ten, na podstawie swojej bazy danych, jest w stanie udzielić bezpośredniej odpowiedzi (9) dotyczącej **client.example.microsoft.com**.
Odpowiedź ta jest z kolei przekazywana (10) przez serwer DNS realizujący powyższe zapytanie w trybie iteracyjnym klientowi, który zainicjował cały proces przesyłając do niego zapytanie rekurencyjne (1).

2. Klient DNS w systemie Linux

Klienta DNS konfigurujemy w systemie Linux z użyciem pliku **/etc/resolv.conf**.
W pliku tym umieszczamy, w kolejnych liniach, adresy serwerów DNS z którymi ma współpracować klient, poprzedzone słowem kluczowym **nameserver**.
Należy pamiętać, iż jedynie pierwszy z podanych serwerów będzie wykorzystywany w zwykłych warunkach – kolejne będą wykorzystywane tylko w sytuacji, gdy żaden z serwerów zdefiniowanych wcześniej nie odpowiedział na żądanie.

W powyższym pliku konfiguracyjnym możemy też zdefiniować domyślną domenę DNS, podając ją po słowie kluczowym **domain**.
Jeśli zamierzamy zdefiniować więcej niż jedną taką domenę, należy posłużyć się słowem kluczowym **serach**, po którym wyliczamy kolejne domeny, rozdzielając je spacjami.

Zmiany w pliku **/etc/resolv.conf** są aktywne od chwili jego zapisania, bez konieczności żadnych dodatkowych działań.

Przykład:

```
nameserver 127.0.0.1
```

```
nameserver 192.168.10.12
domain lab.local
search lab2.local test.pl przyklad.edu
```

3. Serwer BIND – konfiguracja podstawowa

Konfiguracja popularnego serwera DNS wykorzystywanego w systemach Linux (serwer BIND) składa się, w największym uproszczeniu, z 2 typów plików:

- Pliku konfiguracyjnego – odpowiada on za określenie różnorodnych parametrów pracy serwera, stosowanych zabezpieczeń, oraz obsługiwanych przez dany serwer stref.
- Plików bazy danych – zawierają rekordy należące do stref obsługiwanych przez dany serwer.

Przy uruchomieniu serwera, przede wszystkim wczytywany jest plik konfiguracyjny, w którym znajdują się informacje konfiguracyjne niezbędne do uruchomienia serwera.
W pliku konfiguracyjnym zdefiniowane są też wszystkie strefy DNS dla których dany serwer jest autorytatywny, wraz ze wskazaniem gdzie poszukiwać pliku zawierającego informacje o rekordach należących do danej strefy.

3.1 Komentarze

Zarówno w pliku konfiguracyjnym, jak i w plikach bazy danych można stosować komentarze, zaznaczane na jeden z 3 sposobów:

- Poprzedzając komentarz znakiem **#** - jako komentarz traktowany jest tekst znajdujący się po znaku # do końca danej linii
- Poprzedzając komentarz parą znaków **//** - jako komentarz traktowany jest tekst znajdujący się po znakach // do końca danej linii
- Umieszczając komentarz pomiędzy parami znaków **/*** i ***/**

Przykład:

```
# komentarz
// komentarz
/* pierwsza linia komentarza
druga linia komentarza
trzecia linia komentarza */
```

3.2 Uruchamianie serwera BIND

Podczas laboratorium serwer BIND uruchamiamy poleceniem:

```
named -u named -g
```

Spowoduje to uruchomienie serwera z prawami użytkownika „**named**”, co zapobiegnie problemom z uprawnieniami. Ponadto, serwer zostanie uruchomiony jak zwykły program wypisujący w trakcie pracy komunikaty o swoim działaniu na konsole.
Tak uruchomiony serwer może zostać zatrzymany kombinacją **Ctrl+C**.

Serwer odczytuje zmiany konfiguracji przy uruchomieniu. Po wprowadzeniu zmian należy go więc uruchomić ponownie lub użyć polecenia **rndc reload** (patrz 6.1).

Zmieniając konfigurację serwera należy pamiętać, aby serwer DNS posiadał, w swoim folderze roboczym, pełne uprawnienia do zawartych w nim folderów i plików, oraz możliwość tworzenia nowych.

4. Plik konfiguracyjny named.conf

Najważniejszymi z naszego punktu widzenia elementami pliku konfiguracyjnego są wyrażenia **options** i **zone**.
Należy pamiętać, iż każde wyrażenie w pliku konfiguracyjnym serwera BIND musi zostać zakończone średnikiem (**;**) – jego brak jest najczęstszą przyczyną problemów z uruchomieniem serwera po zmianie konfiguracji.

4.1 Polecenie include

Należy pamiętać, iż zarówno w pliku konfiguracyjnym może wystąpić polecenie:
`include "<nazwa_pliku>"`
powodujące wstawienie w miejscu jego występowania zawartości pliku podanego jako parametr.
W ten sposób można podzielić plik konfiguracyjny na kilka osobnych, co może ułatwiać zarządzanie serwerem. Popularną praktyką jest, np. umieszczenie kluczy TSIG w osobnych plikach i dołączenie ich do pliku konfiguracyjnego powyższym poleceniem.

4.2 Listy adresów

Jako wartość wielu opcji konfiguracyjnych (np. listen-on, allow-query, itp.) należy podać listę adresów IP. Format pliku konfiguracyjnego wymaga, aby lista taka była ujęta w nawiasy klamrowe **{ }**, a każdy jej element zakończony był średnikiem **;**. W niektórych przypadkach w ramach listy adresów można podawać także adresy sieci w postaci <adres>/<maska> (np. 10.100.0.0/16). Poszczególne elementy listy mogą znajdować się w osobnych liniach.
Przykładowo lista zawierająca pojedynczy adres IP będzie miała postać { 10.10.0.1; }, lista zawierająca kilka adresów: { 192.168.1.1; 10.10.2.0/24; 204.0.3.1; }.

Na liście adresów mogą również wystąpić wartości specjalne:

- **any** – każdy adres IP,
- **none** – żaden adres IP,
- **localhost** – każdy adres IP komputera na którym uruchomiony jest serwer BIND,
- **localnet** – każdy adres IP należący do sieci, do których należy komputer na którym uruchomiony jest serwer BIND.

Jeśli dany element z listy poprzedzimy znakiem wykrzyknika, oznaczać to będzie usunięcie go z listy. Przykładowo zapis { !192.168.1.254; 192.168.1.0/24; } oznacza sieć 192.168.1.0/24 za wyjątkiem adresu 192.168.1.254.
Elementy listy sprawdzane są po kolei, do momentu znalezienia pasującego – z powyższego względu zapis { 192.168.1.0/24; !192.168.1.254; } oznaczać będzie całą sieć 192.168.1.0/24, gdyż adres 192.168.1.254 zostanie przyjęty jako pasujący do pierwszego jej elementu i drugi element (zakazujący jego użycia) nie będzie już sprawdzany.

W przypadku opcji kontrolujących dostęp (np. allow-update), jako elementy listy adresów można też wykorzystać klucze TSIG (patrz rozdział 7). W takim wypadku uznaje się, iż klient posługujący się takim kluczem należy do danej listy, niezależnie od swojego aktualnego adresu IP.

4.3 Wyrażenie options

Wyrażenie **options** pozwala na ustalenie szeregu parametrów kontrolujących pracę serwera, które umieszczane są w kolejnych liniach pliku konfiguracyjnego, zawartych pomiędzy nawiasami klamrowymi.
Definicja każdego z parametrów musi być zakończona średnikiem (**;**).

```
options {
      <opcja> <wartość>;
      …
};
```

Przykład:

```
options {
      listen-on port 53 { any; };
      listen-on-v6 port 53 { any; };
      directory "/var/named";
      allow-query { any; };
      recursion yes;
      dnssec-enable yes;
      dnssec-validation yes;
      empty-zones-enable no;
      masterfile-format text;
      managed-keys-directory "/var/named/dynamic";
      pid-file "/run/named/named.pid";
      session-keyfile "/var/run/named/session.key";
};
```

Interesujące nas opcje konfiguracyjne to:

- listen-on [port <port>] { <lista_adresów> };
- listen-on-v6 port <port> { <lista_adresów> };
- directory <folder>;
- allow-query { <lista_adresów lub acl> };
- recursion yes|no;
- dnssec-enable yes|no;
- dnssec-validation yes|no;
- empty-zones-enable yes|no;
- pid-file <plik>;
- session-keyfile <plik>;
- managed-keys-directory <folder>;
- masterfile-format test|raw;

4.3.1 listen-on [port <port>] { <lista_adresów> };

4.3.2 listen-on-v6 port <port> { <lista_adresów> };

Powyższe opcje pozwalają na określenie, pod jakimi adresami IP i IPv6 serwer DNS będzie osiągalny dla innych serwerów i klientów DNS. Opcjonalny element port <nr_portu> pozwala na określenie niestandardowego portu TCP i UDP pod którym osiągalny będzie serwer. Standardowo będzie to port 53.

4.3.3 directory <folder>;

Określa folder, w który stanowi domyślną ścieżkę roboczą serwera. Jeśli w innych opcjach konfiguracyjnych podamy ścieżkę względną, to zostanie ona uwzględniona poczynając od powyższego folderu. Dotyczy to również plików bazy danych zawierających rekordy stref DNS obsługiwanych przez serwer.

Np.: w przypadku gdy podaliśmy następujące wartości opcji:

```
directory „/var/named”;
manager-keys-directory “dynamic”;
```

to pełna ścieżka do folderu w którym przechowywane będą dynamicznie uzyskiwane klucze przyjmie postać: /var/named/dynamic

Należy pamiętać aby serwer DNS posiadał, w swoim folderze roboczym, pełne uprawnienia do zawartych w nim folderów i plików, oraz możliwość tworzenia nowych.

W tym celu, w przypadku systemu Linux którym posługujemy się podczas laboratorium, można posłużyć się poleceniem:

```
chown -R named.named <folder>
```

które uczyni serwer DNS właścicielem podanego folderu oraz wszystkich zawartych w nim plików i folderów.

Należy ponadto pamiętać, iż jeśli po wykonaniu takiego polecenia stworzymy w powyższym folderze jakieś nowe pliki, to ich właścicielem będzie najprawdopodobniej (zależy to od konfiguracji systemu) użytkownik jako który byliśmy zalogowani, a nie serwer DNS – w takim przypadku polecenie powyższe należy wykonać ponownie.

4.3.4 allow-query { <lista_adresów> };

Powyższa opcja pozwala na określenie listy adresów źródłowych z których zapytania są przez serwer przetwarzane. Zapytania nadsyłane z adresów nieujętych na liście są odrzucane, a do klienta odsyłany jest stosowny komunikat.

Przykład:

```
allow-query { 10.10.1.1; 192.168.10.0/24; };
allow-query { any; };
```

W tej opcji możliwe jest też wykorzystanie kluczy TSIG (patrz rozdział 7) jako elementów listy adresów, np.:

```
allow-query { key klucz1; };
```

4.3.5 recursion yes|no;

Określa, czy serwer obsługuje zapytania rekurencyjne. Jeśli ich obsługa jest włączona (yes), to klienci mogą przesyłać do niego zapytania rekurencyjne (zwracające ostateczną odpowiedź), a serwer podejmie próbę znalezienia właściwego odwzorowania. Jeśli obsługa jest wyłączona (no), serwer przyjmuje wyłącznie zapytania iteracyjne. Wszystkie zapytania rekurencyjne są odrzucane, a do klienta wysyłany jest odpowiedni komunikat.

4.3.6 dnssec-enable yes|no;

Określa, czy obsługiwane są mechanizmy bezpieczeństwa oparte na podpisach cyfrowych, takie jak:

- TSIG – zabezpieczenia transferów stref i aktualizacji dynamic-update,
- SIG(0) – zabezpieczenia aktualizacji dynamic-update,
- DNSSEC – kryptograficzna weryfikacja informacji przesyłanych klientom DNS w odpowiedzi na przesyłane przez nich zapytania.

4.3.7 empty-zones-enable yes|no;

Umożliwia automatyczne utworzenie na serwerze DNS pustych stref obsługujących zapytania które nie powinny być przekazywane do innych serwerów DNS – dotyczy to w szczególności zapytań odwrotnych (reverse-query) o nazwy hostów o adresach:

- loopback IPv4 i IPv6,
- z przedziału adresów prywatnych IPv4 i IPv6,
- lokalnych łącza IPv6.

Obecność takiej strefy spowoduje, iż serwer samodzielnie odpowie na zapytanie o nazwę hosta o jednym z powyższych adresów komunikatem, że dane odwzorowanie nie istnieje. Jeśli chcemy obsługiwać takie zapytania, należy wyłączyć (lub ograniczyć) automatyczne tworzenie stref pustych.

4.3.8 pid-file <plik>;

Określa nazwę pliku w którym przechowywany jest identyfikator procesu serwera DNS gdy jest on uruchomiony. Umożliwia to innym programom i narzędziom komunikację z powyższym procesem, np. w celu sterowania jego działaniem.

4.3.9 session-keyfile <plik>;

Opcja ta określa gdzie zapisywany jest automatycznie generowany przez serwer DNS klucz TSIG (patrz rozdział 7) o nazwie **local-ddns,** który można wykorzystać aby pozwolić narzędziom uruchamianym na tej samej maszynie (np. nsupdate - patrz 6.3, opcja „-l") co serwer na wykorzystanie powyższego mechanizmu.

4.3.10 masterfile-format text|raw;

Opcja decyduje, w jakim formacie są domyślnie zapisywane pliki bazy danych stref. Format **text** jest formatem tekstowym zrozumiałym dla człowieka, który może być łatwo edytowany z użyciem edytora tekstowego. Format **raw** jest formatem binarnym, który z kolei oferuje większą wydajność pracy serwera.

4.3.11 server <adres_serwera>

Umożliwia określenie specyficznych opcji wykorzystywanych przez konfigurowany serwer DNS do komunikacji z serwerem DNS o podanym adresie <adres_serwera>.
Składnia:

```
serwer <adres_serwera> {
     <opcje>
};
```

Interesujące nas opcje obejmują:

- query-source <adres_ip> - powoduje wysyłanie szelkich żądań do podanego serwera z określonego źródłowego adresu IPv4, jeśli nasz serwer posiada kilka adresów. Może to być przydatne jeśli serwer docelowy wykorzystuje kontrolę dostępu na podstawie adresów źródłowych przychodzących żądań.
- query-source-v6 <adres_ipv6> - jak wyżej, lecz dotyczy adresów IPv6.
- keys { <nazwa_klucza>; } – nakazuje wykorzystywać klucz TSIG o nazwie <nazwa_klucza> w komunikacji z podanym serwerem. Patrz 7.1.

4.4 Wyrażenie zone

Wyrażenie **zone** pozwala na zdefiniowanie strefy DNS bezpośrednio obsługiwanej przez dany serwer, tzn. takiej:

- której baza danych jest na nim przechowywana,
- zapytania dotyczące której są zawsze bezpośrednio udzielane przez danych serwer (bez komunikacji z innymi serwerami DNS).

Serwer DNS przyjmuje, iż ma kompletną wiedzę (jest autorytatywny) dla stref, które są na nim zdefiniowane.

Oznacza to, iż jeśli dany serwer DNS ma zdefiniowaną strefę o określonej nazwie, to udzieli (wykorzystując przypisany do niej plik bazy danych) odpowiedzi na wszystkie zapytania dotyczące nazw kończących się tak jak nazwa strefy. Na przykład, jeśli zdefiniujemy strefę test.pl, to serwer będzie samodzielnie udzielał odpowiedzi na pytania o wszystkie nazwy zakończone na ***test.pl*** – np.***komp.test.pl***, ***test.pl***, ***serwer.gdansk.test.pl***, …

Jeśli zdefiniujemy kilka stref, które obejmują tę samą przestrzeń nazw, np. ***test.pl*** i ***gdansk.test.pl***, to odpowiedź udzielana będzie przy użyciu bazy danych tej z pasujących do zapytania stref, której nazwa jest najdłuższa. Na przykład, w przypadku wspomnianych powyżej stref ***test.pl*** i ***gdansk.test.pl***:

- odpowiedź na zapytanie dotyczące nazw ***www.test.pl*** oraz ***poczta.biuro.test.pl*** będzie udzielona z użyciem bazy danych strefy ***test.pl***,
- odpowiedź na zapytanie dotyczące nazwy ***smtp.gdansk.test.pl*** będzie udzielona z użyciem bazy danych strefy ***gdansk.test.pl***.

Należy pamiętać, iż jeśli w przypisanym do użytej strefy pliku bazy danych nie będzie informacji na temat nazwy której dotyczy zapytanie, serwer odpowie, że nazwa taka nie istnieje.

Składnia:

```
zone <nazwa_strefy> IN {
        <parametry>
};
```

Powyższa definicja strefy może zawierać szereg parametrów, z których wybrane przedstawiono poniżej.

4.4.1 type master|slave;

Parametr wymagany.

Określa, czy dany serwer przechowuje kopię główną (master) czy podrzędne (slave) bazy danych strefy. Należy pamiętać, iż jedynie kopia typu master umożliwia prowadzanie zmian w bazie danych.

4.4.2 file <plik>;

Parametr wymagany.

Określa plik w którym przechowywana jest baza danych strefy. Należy pamiętać, że jeśli użyjemy tu ścieżki względnej, to zostanie ona rozwinięta względem folderu określonego opcją **directory** (parz 4.3.3).

Dla stref typu master powyższy plik musi istnieć. Dla stref typu slave, plik ten, w razie potrzeby, zostanie utworzony automatycznie, a jego treść zostanie pobrana z serwera wskazanego parametrem **masters** (patrz 4.4.3).

4.4.3 masters <lista_adresów>;

Parametr wymagany dla strefy typu slave.
Określa adres serwera DNS przechowującego kopię główną (master) definiowanej strefy. Po uruchomieniu, serwer sprawdzi czy posiadana przez niego kopia bazy danych strefy jest aktualna i jeśli nie, to pobierze ją z serwera o podanym adresie.

4.4.4 allow-transfer <lista_adresów>;

Umożliwia określenie listy adresów źródłowych, z których przyjmowane są żądania transferu interesującej nas strefy.
Domyślnie jest to możliwe z dowolnego adresu (any).
W tej opcji możliwe jest też wykorzystanie kluczy TSIG (patrz rozdział 7) jako elementów listy adresów.

4.4.5 allow-update <lista_adresów>;

Umożliwia określenie listy adresów źródłowych, z których przyjmowane są żądania aktualizacji interesującej nas strefy z użyciem mechanizmu dynamic-update.
Domyślnie nie jest to możliwe z żadnego adresu.
W tej opcji możliwe jest też wykorzystanie kluczy TSIG (patrz rozdział 7).
Opcji **allow-update** nie można stosować, jeśli dla danej strefy zdefiniowano opcje **update-policy** (patrz 7.1.4).

4.4.6 allow-query <lista_adresów>;

Umożliwia określenie listy adresów źródłowych, z których przyjmowane są zapytania dotyczące nazw z interesującej nas strefy.
Domyślnie jest to możliwe z dowolnego adresu (any).
W tej opcji możliwe jest też wykorzystanie kluczy TSIG (patrz rozdział 7) jako elementów listy adresów.

5. Pliki stref

Pliki stref składają się z szeregu wpisów (Resource Records – RRs) mogących zawierać różnorodne informacje – najczęściej jest to odwzorowanie nazwy hosta na jego adres lub odwrotnie.

5.1 Resource Records (RRs)

Pojedynczy wpis składa się z szeregu pól, rozdzielonych znakiem spacji lub tabulatora:

```
<nazwa> <TTL> <klasa> <typ> <dane>
```

Np.:

```
serwer.test.local. 1H IN A 10.10.1.5
```

5.1.1 Nazwa

Nazwa zawiera pełną nazwę DNS rekordu, pozwalającą klientom zgłaszać zapytania, w wyniku których zwrócone zostaną im dane w nim zawarte.
Należy zwrócić uwagę, że nazwa ta musi zostać zakończona kropką „.”. Jeśli tak nie będzie, to na jej końcu zostanie dopisana nazwa strefy lub ciąg znaków zdefiniowany dyrektywą $ORIGIN <nazwa> (patrz niżej).
Jeżeli nazwa zostanie pozostawiona pusta, to w jej miejsce przyjęta zostanie nazwa z poprzedniego rekordu.
W miejscu nazwy można też podać znak @, co oznacza umieszczenie w tym miejscu nazwy strefy.

Dyrektywa $ORIGIN <nazwa> powoduje, iż od miejsca jej występowania, nazwy które nie zostały zakończone kropką i w związku z tym zostałyby uzupełnione o nazwę strefy, będą zamiast tego uzupełniane o ciąg <nazwa>. UWAGA: Ciąg <nazwa> MUSI KOŃCZYĆ SIĘ KROPKĄ.

5.1.2 TTL

Pole opcjonalne. Jeśli nie wystąpi, zostanie przyjęta wartość TTL domyślna dla danej strefy, którą możemy ustawić z użyciem dyrektywy $TTL <czas> (patrz niżej).
Określa czas przez jaki informacja zawarta w danym RR może być przechowywana w pamięci podręcznej klienta. Domyślnie jednostką jest sekunda.
Gdy klient (stacja robocza lub inny serwer DNS) otrzyma od autorytatywnego serwera DNS informację zawartą w danym RR, pytający zapamiętuje ją w swojej pamięci podręcznej. Dzięki temu nie trzeba wysyłać pytania do serwera, gdy informacja zawarta w RR będzie znowu potrzebna. Aby jednak uniknąć problemów związanych z dezaktualizacją takiej informacji (np. zmianą adresu IP odpowiadającego danej nazwie) wprowadzono związany z każdym RR czas życia (TTL – Time To Live), po upływie którego musi on zostać usunięty z pamięci podręcznej, a pytanie do serwera powtórzone w celu uzyskania aktualnej informacji.

Dyrektywa $TTL <czas> powoduje, iż od miejsca jej występowania, dla rekordów dla których nie określono czasu TTL, zostanie przyjęta wartość <czas>.

5.1.3 Klasa

Pole opcjonalne. Jeśli nie wystąpi, to przyjmuje domyślną wartość IN.
Określa ogólne przeznaczenie rekordu. Najpopularniejszą wartością jest IN (od INTERNET), oznaczające, iż wpis wykorzystywany jest przez protokoły wykorzystywane w sieci Internet.

5.1.4 Typ

Typ rekordu szczegółowo określa rodzaj i przeznaczenie zawartej w nim informacji. W chwili obecnej istnieje duża liczba typów rekordów, lecz na potrzeby niniejszego laboratorium skupimy się na poniższych ich typach.

Rekord SOA

Rekord Start-Of-Authority (SOA) musi być pierwszym rekordem w pliku danej strefy, jako że opisuje on jej ogólne parametry oraz zawiera właściwe dla całej strefy wartości domyślne.
Składnia rekordu SOA jest następująca:
<nazwa_strefy> IN SOA <nazwa_serwera> <adres_email> (<nr_seryjny> <refresh> <retry> <expiry> <minimum>)

Pole **nazwa_strefy** musi zawierać nazwę tworzonej strefy, zgodną z tą zdefiniowaną w pliku named.conf.
Pole **nazwa_serwera** zawiera nazwę serwera DNS który przechowuje kopię typu master danej strefy. Obecność tego pola nie zwalnia nas z konieczności umieszczenia w pliku strefy informacji o tym serwerze z użyciem rekordu NS (patrz 0)
Pole **adres_email** zawiera adres email osoby odpowiedzialnej za zarządzanie strefą. W zapisie adresu znak @ należy zastąpić kropką.

Pole **numer_seryjny** zawiera liczbę będącą identyfikatorem konkretnej wersji pliku danych strefy. Po wprowadzeniu do niej jakichkolwiek zmian należy wartość tą zmienić na większą. W przeciwnym wypadku serwery slave nie wykryją wprowadzonych zmian i nie przeprowadzą transferu strefy.
Pole **refresh** określa co ile sekund serwery slave sprawdzają aktualność swojej wersji danych strefy.
Pole **retry** decyduje, co ile sekund serwery slave próbują ponowić proces połączenia się z serwerem master, jeśli poprzednia taka próba zawiodła.
Pole **expiry** określa, po jakim czasie od ostatniego udanego sprawdzenia aktualności danych strefy przez serwer slave uzna on posiadane na jej temat dane za nieaktualne i przestanie ich używać.
Pole **minimum** pozwala określić, jak długo klienci mają przechowywać w pamięci podręcznej odpowiedzi „Nie ma takiego hosta" (NXHOST) dotyczące danej strefy.

Należy pamiętać, iż pola nazwa_strefy, nazwa_serwera i adres_email należy zakończyć kropką, jeśli nie chcemy aby została do nich automatycznie dopisana nazwa strefy (patrz 5.1.1).

Rekord NS

Rekord NS służy do określania nazw serwerów DNS, które są autorytatywne dla określonej strefy, tzn. przechowują jej bazę danych i są w stanie jednoznacznie udzielić odpowiedzi na zapytania jej dotyczące. W każdym pliku strefy należy umieścić rekordy NS wyliczające wszystkie serwery master i slave które przechowują jej bazę danych.
Przykładowo poniższy zapis oznacza, iż serwerami autorytatywnymi dla strefy w której pliku danych go umieściliśmy (@ oznacza nazwę strefy opisywanej przez edytowany plik danych) są serwery DNS pod adresami dns1.test.pl i dns3.lab.pl.

```
@           IN NS dns1.test.pl.
@           IN NS dns3.lab.pl.
```

Należy też pamiętać, iż jeśli nie zakończymy nazwy strefy (nie dotyczy to znaku @) lub nazwy serwera kropką, to zostaną one uzupełnione o nazwę strefy zgodnie z opisem w 5.1.1.

Drugim z zastosowań rekordów NS jest tzw. delegacja strefy, będąca podstawą tworzenia hierarchii usługi DNS. Aby umożliwić usłudze DNS odnalezienie nazw, w bazie danych serwerów obsługujących określoną strefę należy umieścić rekordy NS dotyczące wszystkich stref o nazwach będącej jej rozszerzeniem – np. w strefie test.pl należy umieścić rekordy NS dotyczące stref: gdansk.test.pl, gdynia.test.pl, biuro.gdansk.test.pl, itd.
Dodatkowo, jeśli nazwa serwera DNS dla określonej w powyższy sposób strefy DNS należy do tej strefy (np. adresem serwera DNS dla strefy gdansk.test.pl ma być dns.gdansk.test.pl), to należy dodać tzw. rekord klejący (glue record). Jest to rekord typu A, który pozwoli usłudze DNS określić adres IP serwera DNS którego nazwę można znaleźć w strefie którą dopiero dodajemy, np.:

```
gdansk.test.pl.       IN NS dns.gdansk.test.pl.
dns.gdansk.test.pl.   IN A  192.168.1.20
```

Rekord A

W polu danych zawarty jest adres IPv4 odpowiadający nazwie danego rekordu.
Przykład:

```
serwer                IN A  192.168.1.10
klient.test.pl.       IN A  10.1.1.1
```

Rekord AAAA

W polu danych zawarty jest adres IPv6 odpowiadający nazwie danego rekordu.
Przykład:

```
serwer              IN A  fd00:142:1::ffff
klient.test.pl.     IN A  2001:4070:11:500::de32
```

Rekord PTR i strefy odwrotne

Rekordy PTR występują w tzw. strefach odwrotnych, których celem jest udostępnianie informacji o nazwach DNS odpowiadających adresom IP. Nazwy tych stref są, w celu odróżnienia ich od zwykłych stref, odgórnie zdefiniowane jako:

- dla adresów IPv4 – kończące się na in-addr.arpa.
- dla adresów IPv6 – ip6.arpa.

Adres IP w strefach odwrotnych zapisywany jest z odwróconą kolejnością bajtów, tzn. na przykład adres **192.168.1.2** powinien zostać zapisany jako **2.1.168.192**
W przypadku adresów IPv6, należy podać ich kolejne znaki, począwszy od ostatniego, rozdzielając je kropkami. Niedopuszczalne jest stosowanie skróconego zapisy adresu wykorzystującego znak :: (powodujący dopisanie w miejscu jego użycia ilości zer dopełniającej adres do właściwej długości).

Tak więc całkowita nazwa rekordu mającego zwrócić nazwę hosta odpowiadającą adresowi **192.168.1.2** przyjmie postać **2.1.168.192.in-addr.arpa**.
W przypadku adresu **IPv6 fd00:142:1::ffff** będzie to:
f.f.f.f.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.0.0.0.2.4.1.0.0.0.d.f.ip6.arpa

Ponieważ, zgodnie z zasadami, strefa o nazwie in-addr.arpa powinna zawierać informacje o wszystkich istniejących adresach IPv4, z reguły tworzymy strefy o bardziej szczegółowych nazwach, dotyczące jedynie wycinka adresacji. Np. strefa zawierająca komplet informacji o adresach IPv4 zaczynających się od **192.168.1** będzie miała nazwę: **1.168.192.in-addr.arpa**
W pliku strefy, jeśli w rekordzie PTR jeśli adres IP lub nazwa hosta nie zostanie zakończona kropką, to zostaną one uzupełnieniowe o nazwę strefy w sposób opisany już w pkt. 5.1.1.
Oznacz to, że w strefie **1.168.192.in-addr.arpa** rekord:

```
10                          IN PTR     serwer.test.pl.
```

oraz

```
10.1.168.192.in-addr.arpa. IN PTR     serwer.test.pl.
```

są tożsame, gdyż do **10** w pierwszym z nich zostanie automatycznie dopisana nazwa strefy (ponieważ po **10** nie umieszczono kropki).
Jednocześnie należy zauważyć, iż po nazwie **serwer.test.pl.** kropka została umieszczona w obu przypadkach, gdyż raczej nie jesteśmy zainteresowani, aby dopisać do tej wynikowej nazwy hosta nazwę strefy odwrotnej (czyli w tym przypadku **1.168.192.in-addr.arpa**).

Przykłady:

```
10.1.168.192.in-addr.arpa.                                          IN PTR  serwer.test.pl.
2.3.d.e.0.0.0.0.0.0.0.0.0.0.0.0.0.5.0.1.1.0.0.0.7.0.4.1.0.0.2.ip6.arpa.  IN PTR  klient.test.pl.
```

5.1.5 Dane

Dane które mają zostać zwrócone klientowi pytającemu o dany rekord. Konkretny typ wartości zależy od typu rekordu – np. dla rekordu A będzie to adres IPv4, dla AAAA adres IPv6, dla NS nazwa serwera DNS, dla PTR nazwa hosta, itp.

Pole danych może składać się z wielu pól rozdzielanych spacjami lub znakami tabulacji. Ma to miejsce np. w wypadku typu SRV, gdzie kolejne elementy pola danych zawierają priorytet, wagę, port i adres serwera świadczącego daną usługę.

```
_sip._tcp.example.com. 86400 IN SRV 0 5 5060 sipserver.example.com.
```

5.2 Przykład pliku strefy DNS

```
$TTL 86400
lab142.local.    IN      SOA     fedora.lab142.local. root.lab142.local. (
                         1349
                         1800
                         180
                         604800
                         30 )

                         NS      fedora.lab142.local.
fedora                   A       10.10.0.9

r1                       A       10.10.6.1
r2                       A       10.10.6.2
r3                       A       10.10.6.3
r4                       A       10.10.6.4
```

5.3 Mechanizm aktualizacji dynamic-update i pliki jnl

Mechanizm dynamic-update pozwala klientom aktualizować informacje przechowywane w plika bazy danych serwera DNS. W tym celu klient wysyła specjalne żądanie modyfikacji, do stworzenia którego możemy posłużyć się np. narzędziem **nsupdate** (patrz 6.3).
Kontrola nad uprawnieniami do wprowadzania powyższych modyfikacji zrealizowana jest, między innymi, za pomocą opcji konfiguracyjnych allow-update (patrz 4.4.5) oraz update-policy (patrz 7.1.4). Dzięki nim możliwe jest zastosowanie zarówno kontroli dostępu opartej na prostym określeniu adresów źródłowych które mogą przesyłać żądania modyfikacji, aż do uwierzytelniania żądań z użyciem kluczy mechanizmu TSIG i przydzielania uprawnień do konkretnych rekordów w określonej strefie DNS.

Jeśli wykorzystywany jest mechanizm dynamic-update, to zgłoszone z jego użyciem modyfikacje nie zostają bezpośrednio wprowadzone do plików stref lecz trafiają do plików z rozszerzeniem jnl, tworzonych automatycznie w folderze roboczym serwera. Pomimo, że pliki stref nie są od razu modyfikowane, serwer odpowie na nowe zapytania dotyczące danej strefy uwzględniając poprawki umieszczone tymczasowo w pliku jnl.
Z powyższego względu, nie zaleca się ręcznego modyfikowania plików stref, dla których dopuszczono aktualizacje z wykorzystaniem mechanizmu dynamic-update. Jeśli jest to konieczne, należy przed wprowadzeniem zmian posłużyć się poleceniem **rndc freeze** (patrz 6.1), a po ich wprowadzeniu poleceniem **rndc thaw**.
Jeśli chcemy upewnić się, że pliki stref zostały zaktualizowane z uwzględnieniem wszystkich zgłoszonych żądań aktualizacji, należy posłużyć się poleceniem **rndc sync**. Dodany do niego opcjonalny parametr **-clean**, spowoduje usunięcie plików jnl po wprowadzeniu zawartych w nich poprawek do pliku strefy.

6. Narzędzia dodatkowe

6.1 rndc

Polecenie rndc pozwala na kontrolowanie pracy uruchomionego serwera DNS.
Wywoływane jest w postaci:

```
rndc <polecenie>
```

Przydatne polecenia:

- **freeze [<nazwa_strefy>]** – zablokowanie możliwości wprowadzania zmian do danej strefy z użyciem mechanizmu dynamic-update. Jeśli nie podamy nazwy strefy, dotyczy to wszystkich stref.
- **thaw [<nazwa_strefy>]** – odblokowanie możliwości wprowadzania zmian do danej strefy z użyciem mechanizmu dynamic-update. Jeśli nie podamy nazwy strefy, dotyczy to wszystkich stref.
- **sync [-clean] [<nazwa_strefy>]** – zapisanie wszystkich zmian zgłoszonych z użyciem mechanizmu dynamic-update do pliku danej strefy. Jeśli nie podamy nazwy strefy, dotyczy to wszystkich stref. Parametr -clean nakazuje serwerowi usunąć pliki jnl, po wprowadzeniu zawartych w nich zmian do plików stref (patrz 5.3).
- **status** – wyświetla aktualny stan serwera DNS.
- **reload [<nazwa_strefy>]** – ponowne wczytanie konfiguracji i pliku danych danej strefy. Jeśli nie podamy nazwy strefy ponownie wczytana zostanie cała konfiguracja serwera.
- **tsig-list** – wyświetla listę kluczy TSIG zdefiniowanych na serwerze.

6.2 dig

Polecenie dig jest dość wszechstronnym narzędziem służącym do wysyłania zapytań do serwerów DNS. Pozwala, między innymi, na wysyłanie zapytań rekurencyjnych lub iteracyjnych, do określonych serwerów, o określone typy rekordów, żądań transferu stref itp.

Ogólna składnia polecenia dig to:
dig [@<adres_serwera>] [<parametry>] <zapytanie> [<opcja_zapytania>]

Jeśli zamierzamy wysłać zapytanie do określonego serwera DNS, podajemy jego adres po znaku @ - w przeciwnym przypadku zapytanie zostanie wysłane do domyślnego serwera skonfigurowanego w systemie operacyjnym (w systemie Linux – w pliku /etc/resolv.conf).

Wybrane parametry polecenia dig:

- -4 lub -6 – wysłanie zapytania z użyciem, odpowiednio, IPv4 i IPv6.
- -x – nakazuje traktować treść zapytania jako adres IP którego nazwę DNS należy odnaleźć w strefie odwrotnej.
- -y i -k – pozwalają na uwierzytelnienie wysyłanych zapytań z użyciem mechanizmu TSIG (opis użycia - patrz 7.2).

Zapytanie w poleceniu dig tworzymy podając nazwę DNS o którą pytamy oraz, opcjonalnie, typ zapytania. Jako typ możemy podać:

- any – zwraca rekordy dowolnego typu, odpowiadające danej nazwie,

- <typ_rekordu> – zwraca rekordy podanego typu, odpowiadające danej nazwie,
- axfr – wysyła żądanie pełnego transferu strefy o podanej nazwie,
- ixfr=<numer_seryjny> - wysyła żądanie transferu różnic w obecnej wersji danych podanej strefy w stosunku do wersji o podanym numerze seryjnym.

Jeśli nie podamy typu, domyślnie przyjęty zostanie typ A.

Przykład:
dig klient.test.local
dig @192.168.1.1 -x 10.1.1.15
dig @10.10.1.1 serwer.test.local aaaa
dig @192.168.10.11 test.pl axfr

Opcje zapytania pozwalają na kontrolowanie sposobu przesyłania zapytania i interpretacji odpowiedzi serwera przez program dig:

- +recurse – powoduje wysłanie przez program dig zapytania rekurencyjnego (domyślnie)
- +norecurse – powoduje wysłanie przez program dig zapytania iteracyjnego
- +trace – powoduje ustalenie odpowiedzi przez program dig za pomocą zapytań iteracyjnych i wyświetlenie kolejnych kroków tego procesu.

Przykładowy wynik zapytania `dig @127.0.0.1 kti.eti.pg.gda.pl`:

```
; <<>> DiG 9.8.1-RedHat-9.8.1-2.fc16 <<>> @127.0.0.1 www.eti.pg.gda.pl AAAA
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 7557
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 3, ADDITIONAL: 6

;; QUESTION SECTION:
;www.eti.pg.gda.pl.             IN      AAAA

;; ANSWER SECTION:
www.eti.pg.gda.pl.      68892   IN      AAAA    2001:4070:10:4808::227

;; AUTHORITY SECTION:
eti.pg.gda.pl.          9729    IN      NS      juggernaut.eti.pg.gda.pl.
eti.pg.gda.pl.          9729    IN      NS      smokie.eti.pg.gda.pl.
eti.pg.gda.pl.          9729    IN      NS      ns1.pg.gda.pl.

;; ADDITIONAL SECTION:
ns1.pg.gda.pl.          5292    IN      A       153.19.40.230
ns1.pg.gda.pl.          5292    IN      AAAA    2001:4070:10:4000::230
smokie.eti.pg.gda.pl.   10975   IN      A       153.19.48.100
smokie.eti.pg.gda.pl.   10975   IN      AAAA    2001:4070:10:4800::100
juggernaut.eti.pg.gda.pl. 10975 IN      A       153.19.48.1
juggernaut.eti.pg.gda.pl. 10975 IN      AAAA    2001:4070:10:4800::1

;; Query time: 13 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Wed May 21 14:17:50 2014
;; MSG SIZE  rcvd: 280
```

Nagłówek

W pierwszej linii znajdujemy informacje na temat wersji narzędzia dig, oraz:

- @127.0.0.1 – serwera do którego nakazaliśmy skierować zapytanie,
- www.eti.pg.gda.pl – nazwie o którą pytamy
- AAAA – typie rekordu który nas interesuje.

Ponadto można tu znaleźć wynik zapytania, wskazujący na jego poprawne wykonanie:
status: NOERROR
Niektóre z możliwych wartości tego pola to:

- NOERROR – zapytanie wykonane pomyślnie,
- NXDOMAIN – podana strefa DNS nie istnieje,
- SERVFAIL – błąd w komunikacji z serwerem lub błąd serwera,
- REFUSED – serwer odmówił realizacji żądania.
- BADSIG – błąd weryfikacji wiadomości,
- BADKEY – nieznany klucz,
- BADTIME – niezgodny znacznik czasu, należy sprawdzić, czy zegary klienta i serwera są zsynchronizowane.

Question section

Sekcja zawiera zapytanie na które żądamy odpowiedzi – w tym wypadku rekord AAAA dla nazwy www.eti.pg.gda.pl.

Answer section

Sekcja "Answer" zawiera odpowiedź na przesłane serwerowi DNS żądanie – w tym przypadku widać, że rekord AAAA dla nazwy www.eti.pg.gda.pl ma postać:

```
www.eti.pg.gda.pl.      68892   IN      AAAA    2001:4070:10:4808::227
```

W przypadku gdy serwer nie jest w stanie udzielić odpowiedzi na przesłane żądanie, sekcja „Answer" może nie być obecna.

Authority section

Sekcja "Authority" zawiera informacje o nazwach serwerów DNS autorytatywnych dla strefy której dotyczy żądanie, a więc o tych, które przechowują bazę danych z której udzielono nam odpowiedzi.
W naszym przypadku otrzymaliśmy informacje o nazwach serwerów DNS autorytatywnych dla strefy eti.pg.gda.pl, w której bazie danych znajduje się rekord AAAA dla nazwy www.eti.pg.gda.pl.

Additional section

Sekcja „Additional" zawiera informacje uzupełniające do powyższych sekcji – w typ przypadku rekordy A dotyczące serwerów wskazanych w sekcji „authority" jako będące możliwymi źródłami odpowiedzi na przesłane żądanie, pozwalające na ustalenie ich adresów IP.

Przykładowy wynik zapytania `dig @127.0.0.1 kti.eti.pg.gda.pl +trace`:

```
; <<>> DiG 9.8.1-RedHat-9.8.1-2.fc16 <<>> @127.0.0.1 eti.pg.gda.pl +trace
; (1 server found)
;; global options: +cmd
.                       443221  IN      NS      l.root-servers.net.
.                       443221  IN      NS      e.root-servers.net.
.                       443221  IN      NS      c.root-servers.net.
.                       443221  IN      NS      f.root-servers.net.
```

```
.                       443221  IN      NS      b.root-servers.net.
.                       443221  IN      NS      m.root-servers.net.
.                       443221  IN      NS      g.root-servers.net.
.                       443221  IN      NS      h.root-servers.net.
.                       443221  IN      NS      d.root-servers.net.
.                       443221  IN      NS      i.root-servers.net.
.                       443221  IN      NS      j.root-servers.net.
.                       443221  IN      NS      k.root-servers.net.
.                       443221  IN      NS      a.root-servers.net.
;; Received 512 bytes from 127.0.0.1#53(127.0.0.1) in 11 ms

pl.                     172800  IN      NS      a-dns.pl.
pl.                     172800  IN      NS      c-dns.pl.
pl.                     172800  IN      NS      d-dns.pl.
pl.                     172800  IN      NS      e-dns.pl.
pl.                     172800  IN      NS      f-dns.pl.
pl.                     172800  IN      NS      g-dns.pl.
pl.                     172800  IN      NS      h-dns.pl.
pl.                     172800  IN      NS      i-dns.pl.
;; Received 403 bytes from 199.7.83.42#53(199.7.83.42) in 41 ms

gda.pl.                 86400   IN      NS      bilbo.nask.org.pl.
gda.pl.                 86400   IN      NS      ns1.task.gda.pl.
gda.pl.                 86400   IN      NS      ns2.task.gda.pl.
gda.pl.                 86400   IN      NS      ns.tpnet.pl.
;; Received 156 bytes from 194.181.87.156#53(194.181.87.156) in 20 ms

pg.gda.pl.              3600    IN      NS      sunflower.man.poznan.pl.
pg.gda.pl.              3600    IN      NS      ns2.pg.gda.pl.
pg.gda.pl.              3600    IN      NS      ns1.pg.gda.pl.
;; Received 134 bytes from 2001:4070:1::101#53(2001:4070:1::101) in 2 ms

eti.pg.gda.pl.          86400   IN      A       153.19.55.227
eti.pg.gda.pl.          86400   IN      NS      ns1.pg.gda.pl.
eti.pg.gda.pl.          86400   IN      NS      juggernaut.eti.pg.gda.pl.
eti.pg.gda.pl.          86400   IN      NS      smokie.eti.pg.gda.pl.
;; Received 243 bytes from
2001:4070:10:4000::229#53(2001:4070:10:4000::229) in 1 m
```

Opcja zapytania **+trace** powoduje, że polecenie **dig** ustala odpowiedź na zadane pytanie o rekord A odpowiadający nazwie **eti.pg.gda.pl** samodzielnie wysyłając zapytania iteracyjne.

1. Ponieważ tak nakazaliśmy (@127.0.0.1), pierwsze zapytanie kierowane jest do serwera DNS o adresie **127.0.0.1** (lokalny komputer). Ten nie wie nic na temat poszukiwanej nazwy, więc kieruje nas do serwerów DNS typu **root**, podając ich nazwy.
2. Drugie zapytanie kierowane jest pod adres jednego z otrzymanych w odpowiedzi serwerów **root** (`199.7.83.42`), który z kolei kieruje nas do serwerów obsługujących strefę **pl**.
3. Trzecie zapytanie kierowane jest pod adres jednego z otrzymanych serwerów obsługujących strefę **pl** (`194.181.87.156`) i w odpowiedzi otrzymujemy listę serwerów obsługujących strefę **gda.pl**.
4. Czwarte zapytanie kierowane jest pod adres jednego z otrzymanych serwerów obsługujących strefę **gda.pl** (`2001:4070:1::101`) i w odpowiedzi otrzymujemy listę serwerów obsługujących strefę **pg.gda.pl**.

5. Piąte zapytanie kierowane jest pod adres jednego z otrzymanych serwerów obsługujących strefę **pg.gda.pl** (2001:4070:10:4000::229) i w odpowiedzi otrzymujemy wreszcie informację o zawartości rekordu A dla nazwy **eti.pg.gda.pl**:
   `eti.pg.gda.pl.  86400  IN  A  153.19.55.227`
   Dodatkowo otrzymujemy też informacje o serwerach DNS obsługujących strefę **eti.pg.gda.pl**.

6.3 nsupdate

Narzędzie nsupdate pozwala na przesyłanie do serwera żądań aktualizacji rekordów DNS. W tym celu należy uruchomić je korzystając ze składni:
nsupdate [opcje]
co spowoduje zmianę znaku zachęty na „>", oznaczającą, że jesteśmy w trybie tworzenia żądania aktualizacji z użyciem mechanizmu dynamic-update, które następnie będzie można przesłać do serwera DNS.
Opcje, które można wykorzystać to „-d" oraz „-D", pozwalające na uzyskanie odpowiednio mniejszej oraz większej liczby komunikatów obrazujących przebieg procesu aktualizacji.
Dodatkowo, można wykorzystać opcje „-k" oraz „-y" służące uwierzytelnianiu żądań z użyciem mechanizmu TSIG, opisane w 7.2.
Opcja „-l" pozwala na uruchomienie polecenia nsupdate w trybie komunikacji z serwerem lokalnym (działającym na tej samej maszynie co poleceni nsupdate). W tym trybie nie można zmienić serwera ani kluczy TSIG – ten pierwszy pozostaje ustawiony na lokalną maszynę, a klucz TSIG jest odczytywany z pliku, gdzie przechowywany jest automatycznie wygenerowany przez serwer klucz o nazwie **local-ddns** (patrz 4.3.9).

Wybrane polecenia trybu tworzenia żądania:

- **help** – krótka lista możliwych poleceń,
- **show** – wyświetlenie stworzonego żądania,
- **send** – wysłanie stworzonego żądania,
- **answer** – wyświetlenie odpowiedzi na ostanie wysłane żądanie,
- **server <adres>** - określa serwer DNS do którego przesłane będzie żądanie,
- **update add <nazwa> <ttl> [<klasa>] <typ> <dane>** - żądanie dodania podanego rekordu. Parametry są identyczne z tymi występującymi w pliku bazy danych serwera i opisanymi w 5.1.
- **update delete <nazwa> [<ttl>] [<klasa>] [<typ> [<dane>]]** – żądanie usunięcia rekordu lub wielu rekordów pasujących do podanych parametrów. Parametr <nazwa> jest wymagany. Pozostałe służą dalszemu sprecyzowaniu żądania. Parametr <ttl> jest ignorowany i występuje jedynie dla zachowania jednorodności zapisu rekordu (z poleceniem update add i formatem pliku bazy danych).
- **debug** – powoduje włączenie wyświetlania szczegółowych komunikatów o przebiegu procesu komunikacji z serwerem.

Przykładowy proces aktualizacji:

```
nsupdate
> server 127.0.0.1
> update add test.lab.local 86400 A 192.168.1.50
> update delete test2.lab.local A
> show
Outgoing update query:
;; ->>HEADER<<- opcode: UPDATE, status: NOERROR, id:      0
```

```
;; flags:; ZONE: 0, PREREQ: 0, UPDATE: 0, ADDITIONAL: 0
;; UPDATE SECTION:
test.lab.local.         86400   IN      A       192.168.1.50
test2.lab.local.        0       ANY     A
> send
> answer
Answer:
;; ->>HEADER<<- opcode: UPDATE, status: NOERROR, id:  11142
;; flags: qr ra; ZONE: 1, PREREQ: 0, UPDATE: 0, ADDITIONAL: 0
;; ZONE SECTION:
;lab.local.                    IN      SOA
> quit
```

6.4 dnssec-keygen

Polecenie pozwala na wygenerowanie kluczy wykorzystywanych przez różnorodne mechanizmy zabezpieczeń usługi DNS w tym TSIG oraz DNSSEC.

Polecenie wywołujemy posługując się następującą składnią

```
dnssec-keygen -a <algorytm> -b <długość> [-f KSK]
-n <zastosowanie> <nazwa_klucza>
```

gdzie parametr:

- -a <algorytm> – określa algorytm w którym wykorzystywany będzie klucz, np.: HMAC-MD5, HMAC-SHA1, itp.
- -b <długość> - określa długość klucza w bitach,
- -n <zastosowanie> <nazwa> - określa zastosowanie klucza oraz jego nazwę. Zastosowania to np.:
    - HOST – klucz służący uwierzytelnianiu urządzenia, używany np. w przypadku mechanizmu TSIG,
    - ZONE – klucz służący zabezpieczeniu danych strefy z wykorzystaniem mechanizmu DNSSEC.
- -f KSK - parametr opcjonalny, używany do wygenerowania klucza typu Key Signing Key (KSK) w przypadku zastosowania typu ZONE. Bez tego parametru, dla zastosowania ZONE wygenerowany zostanie klucz Zone Signing Key (ZSK).

Powyższa składnia spowoduje wypisanie nazwy w formacie:
K<nazwa_klucza>.+<algid>+<fingerprint>
Gdzie:

- <nazwa_klucza> – nazwa utworzonego klucza,
- <algid> – identyfikator algorytmu (w postaci cyfrowego numeru algorytmu),
- <fingerprint> – identyfikator („odcisk palca”) klucza.

Zostaną też utworzone 2 pliki o nazwach opartych na powyżej opisanej:

- K<nazwa_klucza>.+<algid>+<fingerprint>.public – zawiera wygenerowany klucz symetryczny lub też klucz publiczny w przypadku algorytmów asymetrycznych,
- K<nazwa_klucza>.+<algid>+<fingerprint>.key – zawiera wygenerowany klucz symetryczny lub też klucz prywatny w przypadku algorytmów asymetrycznych.

7. Mechanizm Transaction SIGnature (TSIG)

Mechanizm TSIG (zdefiniowany w RFC 2845) pozwala na zweryfikowanie tożsamości nadawcy oraz ochronę integralności przesyłanych wiadomości. Możliwe zastosowania tego mechanizmu obejmują np.:

- kontrolę dostępu do funkcji takich jak transfery stref czy uaktualnienia dynamic-update,
- ochronę danych przesyłanych w ramach powyższych operacji przed modyfikacją.

Zabezpieczenie TSIG to bazuje na znajomości tego samego tajnego klucza przez obie komunikujące się strony. Klucz ten używany jest w procesie wyliczania sumy kontrolnej przesyłanej wiadomości z użyciem jednokierunkowych funkcji skrótu takich jak np.: hmac-md5, hmac-sha1, hmac-sha224, hmac-sha256, hmac-sha384 czy hmac-sha512.
W celu ochrony przed atakami typu replay (gdzie atakujący wysyła wcześniej przechwyconą próbkę ruchu), powyższa suma kontrolna obejmuje również znacznik czasowy – oznacza to, że zegary obu komunikujących się stron muszą być zsynchronizowane.
Powyższa suma kontrolna jest dołączana do przesyłanych wiadomości DNS wraz z nazwą klucza użytego do jej stworzenia. Odbiorca, dysponujący kluczem o tej samej nazwie, weryfikuje z jego użyciem poprawność sumy kontrolnej. Jeśli jest ona poprawna, oznacz to, że:

- nadawca dysponował poprawnym kluczem o określonej nazwie – co pozwala potwierdzić jego tożsamość,
- wiadomość nie została zmodyfikowana i została wygenerowana w niedalekiej przeszłości.

Nazwa klucza musi unikalnie identyfikować dany klucz w zbiorze kluczy wykorzystywanych przez określone urządzenie. Zaleca się, aby pojedynczy klucz wykorzystywać w celu ochrony komunikacji wyłącznie pojedynczej pary urządzeń, a nazwy kluczy ustalać w sposób odzwierciedlający ich nazwy, np.:

- <id>.A.B.test.pl. – dla hostów A.test.pl i B.test.pl.
- <id>.A.test.pl.<id>B.lab.pl. – dla hostów A.test.pl i B.lab.pl.

Wartość <id> w powyższych przykładach odzwierciedla możliwość, iż dana para urządzeń wykorzystuje do komunikacji ze sobą kilka kluczy (np. jeden dla aktualizacji dynamic-update, a drugi dla transferów stref).

7.1 Konfiguracja dla serwera BIND

W celu wykorzystania mechanizmu TSIG na serwerze BIND, należy:

- Dodać definicje kluczy do pliku konfiguracyjnego serwera (**named.conf**).
- Umieścić odwołania do zdefiniowanych kluczy na listach kontrolujących dostęp do odpowiednich funkcji serwera np. allow-update, allow-transfer, itp.
- Dodać do konfiguracji serwera definicję sposobu komunikacji z innymi serwerami wymagającymi uwierzytelniania TSIG.

7.1.1 Definicja klucza

Definicje kluczy można umieścić bezpośrednio w pliku konfiguracyjnym serwera (**named.conf**), lub w osobnym pliku który włączamy do pliku konfiguracyjnego poleceniem **include**.

Definicja klucza ma postać:

```
key <nazwa_klucza> {
    algorithm hmac-md5;
    secret "<treść_klucza>";
```

```
};
```
Nazwa klucza powinna spełniać wymagania opisane w rozdziale 7, a w szczególności MUSI być taka sama na obu komunikujących się urządzeniach.
Jako algorytm używany do tworzenia sumy kontrolnej wykorzystujemy hmac-md5, którego obsługa jest wymagana dla serwerów obsługujących mechanizm TSIG.
Treść klucza zapisana jest w formacie Base64 i powinien mieć dokładnie 128 bitów. W celu jego wygenerowania najlepiej użyć polecenia:
```
dnssec-keygen -a HMAC-MD5 -b 128 -n HOST <nazwa_klucza>
```
Dokładny opis parametrów powyższego polecenia – patrz 6.4.

Intersujący nas klucz należy odczytać z zawartości utworzonych przez polecenie dnssec-keygen plików i umieścić w pliku konfiguracyjnym serwera, zgodnie z opisaną wyżej składnią.

Należy też pamiętać, iż jeśli wykorzystano opcję pliku konfiguracyjnego ***session-key <plik>*** (patrz 4.3.9), to przy każdym uruchomieniu serwera w podanym pliku zostanie zapisany nowo wygenerowany klucz TSIG o nazwie **local-ddns**, któremu można zezwolić dostęp do różnych funkcji serwera tymi samymi metodami co kluczom wygenerowanym klasycznie (patrz niżej).

7.1.2 Zastosowanie w celu kontroli dostępu do serwera

W poleceniach kontrolujących dostęp do rozmaitych funkcji serwera, jak np.:

- allow-query
- allow-update
- allow-transfer

można, w miejsce listy dozwolonych adresów źródłowych, podać klucze TSIG upoważniające do uzyskania dostępu do danej funkcji.

Klucz podajemy w takim przypadku, posługując się składnią:
```
key <nazwa_klucza>;
```
czyli np.:
```
allow-transfer { key dns.test.pl; };
allow-update { key local-ddns; key dns.test.pl; };
```

7.1.3 Zastosowanie w celu komunikacji z innymi serwerami

W celu nakazania serwerowi użycia określonego (zdefiniowanego we wcześniej określony sposób) klucza TSIG w komunikacji z innym serwerem, należy użyć wyrażenia server w pliku konfiguracyjnym. Polecenie to zostało opisane w 4.3.11.

Przykładowa konfiguracja nakazująca naszemu serwerowi na użycie klucza o nazwie test1 w komunikacji z serwerem DNS o adresie 10.1.1.3 wygląda następująco:
```
server 10.1.1.3 {
     keys { test1; };
};
```

7.1.4 Szczegółowa kontrola dostępu w mechanizmie dynamic-update - update-policy

W przypadku definicji stref, możliwe jest użycie opcji konfiguracyjnej **update-policy**. Pozwala ona na szczegółową kontrolę (w tym opartą o wykorzystanie kluczy TSIG) nad uprawnieniami związanymi z aktualizacją danych strefy z wykorzystaniem mechanizmu dynamic-update. Nie można wykorzystywać opcji **update-policy**, jeśli dla danej strefy zdefiniowano już opcję **allow-update**.

Składnia:

```
update-policy {
      <parametry>
};
```

Parametry opcji update-policy mają postać szeregu reguł zdefiniowanych wg następującego wzorca:
`<działanie> <klucz_TSIG > <typ_porównania> <nazwa> [<typ_rekordu>] ;`

Gdzie:

- <działanie> – przyjmuje wartości grant lub deny. Grant nakazuje zezwalać na operację opisywaną przez dany rekord, podczas gdy denny zabrania takiej operacji.
- <klucz_TSIG> – nazwa klucza TSIG użytego do podpisania żądania aktualizacji. Możliwe jest użycie znaków maski w rodzaju *.
- <typ_porównania> - określa sposób analizy treści żądania aktualizacji, w celu określenia czy pasuje do danej reguły. Wybrane wartości przedstawiono poniżej:
    - name – nazwa przesłana w żądaniu aktualizacji musi dokładnie odpowiadać parametrowi <nazwa> reguły,
    - subdomain – nazwa przesłana w żądaniu aktualizacji musi dokładnie odpowiadać parametrowi <nazwa> reguły lub być jego poddomeną,
    - wildcard – nazwa przesłana w żądaniu aktualizacji musi odpowiadać parametrowi <nazwa> w którym dopuszczalne jest użycie znaków maski,
    - self – nazwa przesłana w żądaniu aktualizacji musi dokładnie odpowiadać nazwie użytego klucza TSIG. Pole <nazwa> nie jest używane, lecz powinno mieć wartość taką samą jak pole <klucz_TSIG>,
    - selfsub – nazwa przesłana w żądaniu aktualizacji musi dokładnie odpowiadać nazwie użytego klucza TSIG lub być poddomeną tej nazwy. Pole <nazwa> nie jest używane, lecz powinno mieć wartość taką samą jak pole <klucz_TSIG>.
- <nazwa> - nazwa rekordu, interpretowana w zależności od wartości pola <typ_porównania>. Możliwe jest użycie znaków maski w rodzaju *.
- <typ_rekordu> - parametr opcjonalny, określa jakich typów rekordów dotyczy reguła. Dopuszczalne wartości obejmują typy rekordów DNS oraz ANY (każdy typ). Jeśli nie zostanie podany, oznacza każdy typ rekordu poza SIG, NS, SOA, and NXT.

Przykład:

```
zone „lab.local” IN {
      type master;
      file „lab.local.zone”;
      update-policy {
            grant klucz1 name test.lab.local A;
            deny klucz2 name komp1.lab.local;
            grant klucz2 subdomain lab.local;
      };
};
```

7.2 Zastosowanie w narzędziach klienckich

7.2.1 nsupdate

Opisane wcześniej (6.3) narzędzie nsupdate umożliwia użycie kluczy TSIG do uwierzytelniania przesyłanych do serwera DNS żądań aktualizacji.

```
nsupdate -k <plik>
```

Powoduje pobranie i wykorzystanie przez narzędzie nsupdate klucza TSIG z pliku o podanej nazwie. Obsługiwane są 2 formaty pliku: format uzyskany w wyniku działania polecenia dnssec-keygen (6.4) oraz format zapisu właściwy dla pliku konfiguracyjnego serwera DNS (7.1.1).

```
nsupdate -y <nazwa_klucza>:<treść_klucza>
```

Powoduje wykorzystanie przez narzędzie nsupdate klucza TSIG o nazwie podanej jako parametr <nazwa_klucza> i treści podanej jako parametr <treść_klucza>.
Na przykład:

```
nsupdate -y klient.test.pl:skrKc4Twy/cIgIykQu7JZA==
```

7.2.2 dig

Podobnie jak polecenie **nsupdate**, polecenie **dig** może wykorzystywać klucze TSIG do uwierzytelnienia zapytań i odpowiedzi wymienianych z serwerem DNS.
Używane w tym celu parametry linii poleceń są w jego wypadku takie same jak w wypadku polecenia **nsupdate**.

Mechanizm widoków (Views)

Mechanizm widoków pozwala na zróżnicowanie sposobu przetwarzania żądania przez serwer DNS w zależności od (między innymi) adresu źródłowego z którego zostało ono przesłane lub klucza TSIG którym zostało zabezpieczone.
Uproszczoną składanię definicji nowego widoku przedstawiono poniżej:

```
view <nazwa> {
      match-clients { <lista_adresów> };
      match-destinations { <lista_adresów> };
      match-recursive-only yes|no ;

      [ <opcje> ]
      [ <definicje_stref> ]
};
```

O zakwalifikowaniu żądania do określonego widoku decydują poniższe wyrażenia. Żądanie zostanie zakwalifikowane do danego widoku, jeśli spełnia warunki KAŻDEGO z nich – możliwe jest natomiast zdefiniowanie tylko wybranych z poniższych wyrażeń:

- match-clients { <lista_adresów> }; - do widoku kwalifikowane są żądania przychodzące z adresów źródłowych z podanej listy adresów.
- match-destinations { <lista_adresów> }; - do widoku kwalifikowane są żądania wysłane na adresy z podanej listy. Przydatne gdy nasz serwer posiada kilka adresów IP.
- match-recursive-only yes|no; - określa czy do widoku kwalifikowane są tylko żądania rekurencyjne (yes) czy wszystkie (no).

Po otrzymaniu żądania serwer sprawdza czy pasuje ono do któregoś ze zdefiniowanych widoków, w kolejności ich zdefiniowania w pliku konfiguracyjnym. Jeśli tak, to używany jest pierwszy pasujący widok. Jeśli żądanie nie pasuje do żadnego z nich, używany jest tzw. widok domyślny, czyli konfiguracja zdefiniowana poza blokami view { … };.

Mówiąc w pewnym uproszczeniu, w blokach view { … }; możemy niektóre z opcji konfiguracyjnych właściwych dla bloku options { … } (patrz 4.3), a także definiować strefy DNS (wyrażenie zone - 4.4), które będą widoczne wyłącznie dla urządzeń zakwalifikowanych do danego widoku.
Dzięki temu możliwe jest zdefiniowanie stref o ograniczonym dostępie (zdefiniowanych tylko w określonym widoku), lub też zróżnicowania sposobu dostępu do stref, przez zdefiniowanie tej samej strefy (ta sama nazwa, ten sam plik bazy danych, lecz różne opcje) w kilku widokach.

Przykład definicji widoku:

```
view widok1 {
        match-clients { 192.168.3.0/24; };

        zone „test.lab.local” IN {
                type master;
                file “test.lab.local.zone”;
                allow-update { 192.168.5.0/24; };
        }
}
```

Widok o nazwie **widok1**, dostępny dla klientów którzy o adresach z sieci 192.168.3.0/24. Klienci używający tego widoku mają, jako jedyni (chyba że strefa ta została dodatkowo zdefiniowana również w innym miejscu pliku konfiguracyjnego), dostęp do strefy o nazwie „test.lab.local”.
Opcja allow-update {192.168.5.0/24; }, nie ma w tym przypadku sensu, ponieważ zezwala na aktualizację danych strefu klientom z sieci 192.168.5.0/24, podczas gdy jest zdefiniowana w widoku, z którego korzystają tylko klienci z sieci 192.168.3.0/24.

```
view widok2 {
        match-clients { key klucz1; };

        recursion no;

        zone „test.lab.local” IN {
                type master;
                file “test.lab.local.zone”;
        }
}
```

Widok o nazwie **widok2**, dostępny dla klientów którzy użyli klucza o nazwie klucz1 (zdefiniowanego wcześniej w pliku konfiguracyjnym z użyciem polecenia key – patrz 7.1.1). Klienci używający tego widoku nie mogą przesyłać zapytań rekurencyjnych lecz mają, jako jedyni (chyba że strefa ta została dodatkowo zdefiniowana również w innym miejscu pliku konfiguracyjnego), dostęp do strefy o nazwie „test.lab.local”.