

# Konfiguracja serwera DNS (Bind9) w systemie Linux Debian 11.

Marian Dorosz

---

## Spis treści

<b>1</b>	<b>Instalacja Bind9</b>	<b>4</b>
1.1	Pakiet instalacyjny . . . . .	4
<b>2</b>	<b>Generowanie klucza TSIG</b>	<b>4</b>
2.1	Czym jest TSIG? . . . . .	4
2.2	Generowanie klucza TSIG. . . . .	4
<b>3</b>	<b>Konfiguracja serwera DNS, rekordów RR oraz stref</b>	<b>5</b>
3.1	Konfiguracja pliku named.conf . . . . .	5
3.1.1	Konfiguracja ACL poleceniem: <code>acl nazwa_acl</code> . . . . .	5
3.1.2	Dodawanie klucza TSIG do dynamicznej aktualizacji . . . . .	5
3.1.3	Dodawanie kanału komunikacyjnego do zarządzania BIND9 z poziomu komputera lokalnego z wykorzystaniem RNDCC (controls { ... }) . . . . .	5
3.2	Konfiguracja pliku named.conf.options . . . . .	6
3.2.1	Konfiguracja portów i adresów, którymi serwery DNS będą się wymieniać informacjami . . . . .	6
3.2.2	Konfiguracja serwera, do którego mają być przesyłane nierozwiązane zapytania . . . . .	6
3.2.3	Nasłuchiwanie tylko na lokalnych interfejsach . . . . .	7
3.2.4	Zablokowanie wymiany stref . . . . .	7
3.2.5	<code>allow-query {adresy_wewnetrzne;};</code> . . . . .	7
3.2.6	<code>allow-recursion</code> . . . . .	8
3.3	Konfiguracja pliku named.conf.local . . . . .	8
3.3.1	<code>zone "test.pl"</code> . . . . .	8
3.3.2	<code>zone 0.168.192.in-addr.arpa</code> . . . . .	9
3.4	Konfiguracja rekordów <i>Resource Records</i> . . . . .	9
3.4.1	Czym są rekordy RR? . . . . .	10
3.4.2	SOA - Start of authority record . . . . .	10
3.4.3	Rekord NS . . . . .	10
3.4.4	Rekord A . . . . .	10
3.4.5	Rekord CNAME . . . . .	11
3.4.6	Rekord MX . . . . .	11
3.4.7	Rekord PTR . . . . .	11
3.5	Test przy pomocy narzędzia dig. . . . .	12

---

## Spis rysunków

1	Generowanie klucza TSIG . . . . .	5
2	Gotowy plik named.conf . . . . .	6
3	Skonfigurowany plik named.conf.options . . . . .	7
4	Skonfigurowany plik named.conf.local . . . . .	8
5	Rekordy RR dla strefy test.pl . . . . .	9
6	Rekordy RR dla strefy 0.168.192.in-addr.arpa . . . . .	10
7	Wynik komendy Dig dla domenty test.pl . . . . .	12
8	Wynik komendy Dig dla dns1.test.pl . . . . .	12

---

# 1 Instalacja Bind9

## 1.1 Pakiet instalacyjny

Instalacja aplikacji Bind9 na serwerze wymaga skorzystania z polecenia **apt-get install bind9**. Polecenie to zainstaluje aplikację w systemie. Po zakończeniu procesu instalacji można przystąpić bezpośrednio do konfiguracji.

# 2 Generowanie klucza TSIG

## 2.1 Czym jest TSIG?

TSIG (transaction signature) to protokół umożliwiający aktualizację bazy danych DNS w bezpieczny sposób. Najczęściej wykorzystuje się go do aktualizacji dynamicznego DNS, bądź serwerów DNS działających w trybie *slave*. TSIG wykorzystuje klucze typu **shared secret** (w dużym skrócie chodzi o to, że komputery, które biorą udział w komunikacji znają klucz shared secret) w celu szyfrowania (w jedną stronę) wymiany informacji. Należy tutaj rozróżnić, że aktualizacja serwerów DNS (ich konfiguracji) różni się od wysłania zapytania do serwera (tzw. *DNS query*).

## 2.2 Generowanie klucza TSIG.

Aby wygenerować klucz można skorzystać z polecenia **tsig-keygen** (można także użyć **dnssec-keygen**). Wygenerowany klucz najlepiej zapisać w nowym pliku, który będzie dołączany do konfiguracji aplikacji bind9.

---

```
root@debian:/etc/bind# cp rndc.key ns-test.pl_rndc-key
root@debian:/etc/bind# tsig-keygen -a HMAC-MD5 test.pl > ns-test.pl_rndc-key
root@debian:/etc/bind# cat ns-test.pl_rndc-key
key "test.pl" {
    algorithm hmac-md5;
    secret "Rze/CrVYYdYBxmuaVnF2WA==";
};
root@debian:/etc/bind# _
```

Rysunek 1: Generowanie klucza TSIG

## 3 Konfiguracja serwera DNS, rekordów RR oraz stref

### 3.1 Konfiguracja pliku `named.conf`

Plik *named.conf* jest głównym plikiem konfiguracyjnym serwera DNS.

#### 3.1.1 Konfiguracja ACL poleceniem: `acl nazwa_acl`

ACL to lista adresów IP, które będą mogły podłączyć się do serwera DNS i go konfigurować.

#### 3.1.2 Dodawanie klucza TSIG do dynamicznej aktualizacji

Jest to załączenie pliku z kluczem TSIG przy pomocy dyrektywy `include`.

#### 3.1.3 Dodawanie kanału komunikacyjnego do zarządzania BIND9 z poziomu komputera lokalnego z wykorzystaniem RNDC (`controls { ... }`)

Zezwolenie na połączenie się z serwerem DNS przy pomocy RNDC z komputera o adresie 127.0.0.1 przy użyciu portu 953.

---

```
GNU nano 5.4 /etc/bind/named.conf
// This is the primary configuration file for the BIND DNS server named.
//
// Please read /usr/share/doc/bind9/README.Debian.gz for information on the
// structure of BIND configuration files in Debian, *BEFORE* you customize
// this configuration file.
//
// If you are just adding zones, please do that in /etc/bind/named.conf.local

//Konfiguracja ACL
acl adresy_wewnetrzne {127.0.0.0/8; 192.168.0.0/24; };

//Load options
include "/etc/bind/named.conf.options";

//Dodawanie klucza TSIG do dynamicznej aktualizacji
include "/etc/bind/ns-test.pl_rndc-key";

//Dodawanie kanalu komunikacyjnego do zarzadzania BIND9 z poziomu komputera lokalnego
controls {
    inet 127.0.0.1 port 953 allow { 127.0.0.1; };
};

// Dodawanie strefy ".", ktora jest podpowiedzia gdzie sa informacje o serwerach root
//zone "." {
//    type hint;
//    file "/etc/bind/db.root";
//};

include "/etc/bind/named.conf.local";
include "/etc/bind/named.conf.default-zones";
```

Rysunek 2: Gotowy plik named.conf

## 3.2 Konfiguracja pliku named.conf.options

Plik *named.conf.options* zawiera wszystkie opcje konfiguracyjne dla serwera DNS.

### 3.2.1 Konfiguracja portów i adresów, którymi serwery DNS będą się wymieniać informacjami

Jak widać na powyższym obrazku poleceniem **query-source address \* port \*** zezwalamy serwerowi DNS na komunikację z serwerami o dowolnych adresach na dowolnych portach.

### 3.2.2 Konfiguracja serwera, do którego mają być przesyłane nierozwiązane zapytania

Opcje **forward only**; oraz **forwarders{...}** są informacją dla serwera, gdzie przesłać nierozwiązane zapytanie.

```
GNU nano 5.4 /etc/bind/named.conf.options
//=====
dnsssec-validation auto;

//Konfiguracja portow i adresow ktorymi serwery DNS beda sie wymieniac informacjami
query-source address * port *;

// Jezeli ten serwer nie bedzie znac odpowiedzi na zapytanie to odeslij informacje do
//serwera 192.168.1.1
forward only;
forwarders{192.168.1.1};

auth-nxdomain no;

// Wylaczenie skanowania interfejsow zeby zapobiec niechcianemu przerwaniu nasluchu
interface-interval 0;

// Nasluchiwanie tylko na lokalnych interfejsach IPv4
listen-on-v6 {none;};
listen-on {127.0.0.1; 192.168.0.1;};

// Zabron wymiany stref
allow-transfer {none;};

// Akceptuj zapytania tylko z sieci wew.
allow-query {adresy_wewnetrzne;};

// Zezwol na rekurencyjne wysylanie zapytan do hostow wew.
allow-recursion {adresy_wewnetrzne;};

// Nie tworzy publicznej wersji BINDa
version none;

};
```

Rysunek 3: Skonfigurowany plik named.conf.options

### 3.2.3 Nasłuchiwanie tylko na lokalnych interfejsach

Korzystając z poleceń **listen-on-v6: none;** oraz **listen-on {...}** powoduje, że serwer DNS nie będzie odpowiadał na zapytania pochodzące z adresów IPv6 oraz na zapytania przychodzące na adres inny niż wymieniony w nawiasach klamrowych.

### 3.2.4 Zablokowanie wymiany stref

Poleceniem **allow-transfer { none };** powoduje, że serwer nie będzie udostępniać informacji o strefach innym serwerom.

### 3.2.5 allow-query {adresy\_wewnetrzne};

Polecenie to powoduje, że zapytania do serwera DNS będą mogły pochodzić z podsieci 127.0.0.1/8 (localhost) oraz 192.168.0.0/24. Wynika to z konfiguracji pliku z obrazu 2.

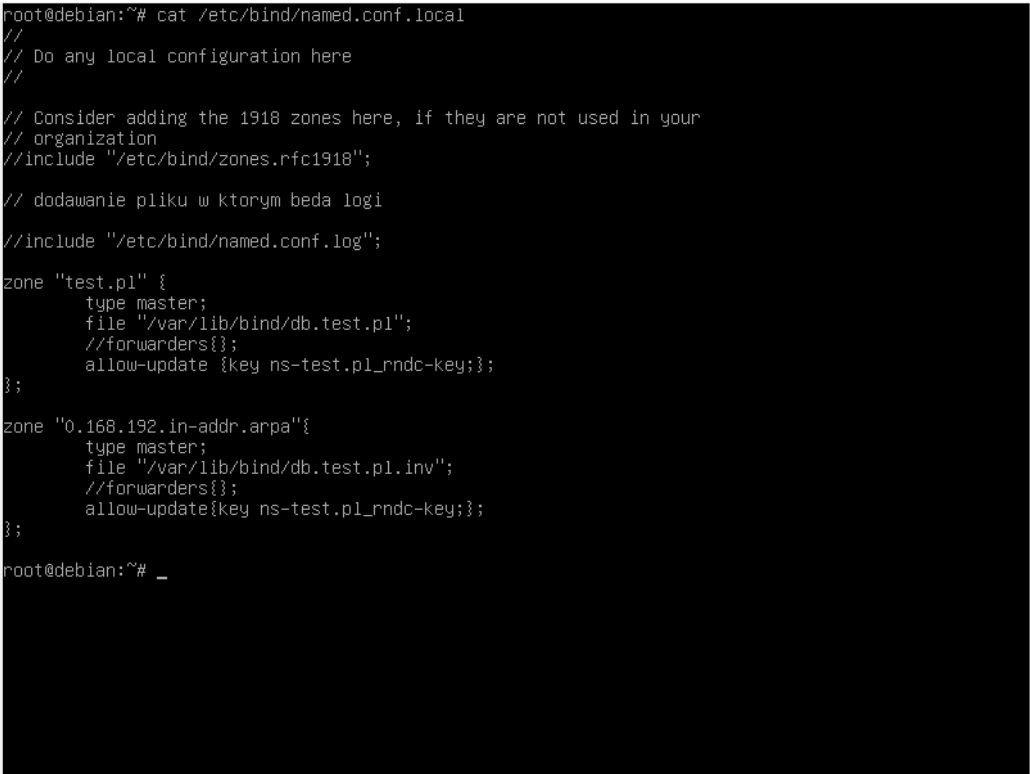
---

### 3.2.6 allow-recursion

Pozwala na wysyłanie przez serwer zapytań do hostów pochodzących z dodanych ACL.

## 3.3 Konfiguracja pliku named.conf.local

W tym pliku konfiguruje się lokalne strefy DNS. W ustawieniach strefy należy dodać informację o typie strefy, o serwerach działających jako *forwarderzy* oraz o bazach danych DNS i plikach z kluczami, które pozwalają na aktualizację wcześniej wspomnianych baz.



```
root@debian:~# cat /etc/bind/named.conf.local
//
// Do any local configuration here
//
// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";

// dodawanie pliku w którym beda logi
//include "/etc/bind/named.conf.log";

zone "test.pl" {
    type master;
    file "/var/lib/bind/db.test.pl";
    //forwarders{};
    allow-update {key ns-test.pl_rndc-key;};
};

zone "0.168.192.in-addr.arpa"{
    type master;
    file "/var/lib/bind/db.test.pl.inv";
    //forwarders{};
    allow-update{key ns-test.pl_rndc-key;};
};

root@debian:~# _
```

Rysunek 4: Skonfigurowany plik named.conf.local

### 3.3.1 zone "test.pl"

W ten sposób została dodana strefa test.pl. Jej typ to *master*, a klucz, który zezwala na aktualizację tej strefy znajduje się w pliku



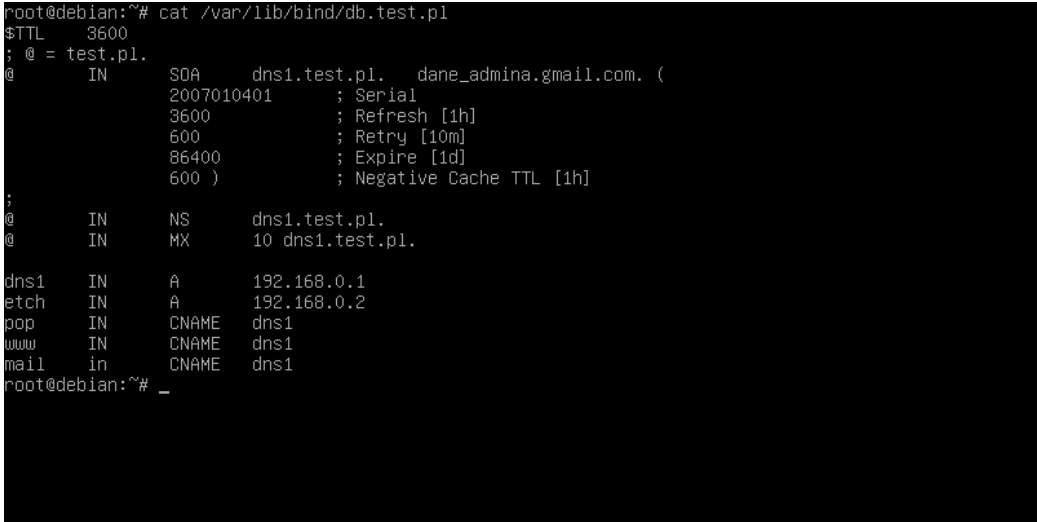
---

**"ns-test.pl\_rndc-key";**. Plik z rekordami RR tej strefy ustawiony param-  
terem **file** to  
**"/var/lib/bind/db.test.pl"**.

### 3.3.2 zone 0.168.192.in-addr.arpa

W ten sposób dodaje się strefę ARPA (tzw. *odwrotny DNS*). Tak jak wcześniej dodana strefa test.pl strefa ARPA jest typu master, jej plik z rekordami RR to **"/var/lib/bind/db.test.pl.inv"**; Nie posiada ona żadnych serwerów działających jako *forwarderzy*. Klucz umożliwiający aktualizację strefy to plik **"ns-test.pl\_rndc-key"**;

## 3.4 Konfiguracja rekordów *Resource Records*



```
root@debian:~# cat /var/lib/bind/db.test.pl
$TTL      3600
; @ = test.pl.
@         IN      SOA      dns1.test.pl.  dane_admina.gmail.com. (
        2007010401      ; Serial
        3600            ; Refresh [1h]
        600             ; Retry [10m]
        86400           ; Expire [1d]
        600 )           ; Negative Cache TTL [1h]
;
@         IN      NS       dns1.test.pl.
@         IN      MX       10 dns1.test.pl.

dns1      IN      A        192.168.0.1
etch      IN      A        192.168.0.2
pop       IN      CNAME    dns1
www       IN      CNAME    dns1
mail      in      CNAME    dns1
root@debian:~# _
```

Rysunek 5: Rekordy RR dla strefy test.pl

---

```
root@debian:~# cat /var/lib/bind/db.test.pl.inv
@      IN      SOA      dns1.test.pl. dane_admina@gmail.com. (
        2007010401
        3600
        600
        86400
        600 )
;
@      IN      NS       dns1.test.pl.
1      IN      PTR      dns1.test.pl.
2      IN      PTR      etch.test.pl.
root@debian:~#
```

Rysunek 6: Rekordy RR dla strefy 0.168.192.in-addr.arpa

### 3.4.1 Czym są rekordy RR?

Rekordy RR oznaczają jaki typ informacji przechowuje dana strefa DNS. Każdy rekord ma swój typ, czas, po którym wygasa oraz informacje specyficzne dla samego siebie.

### 3.4.2 SOA - Start of authority record

Rekord ten przechowuje autorytatywne informacje o strefie DNS, włączając w to główny serwer rozpoznawania nazw, email administratora, numer seryjny strefy oraz kilka liczników czasu, które powiązane są z odświeżaniem informacji o strefie. Liczniki te są informacjami dla serwerów zapasowych, które mają synchronizować się z głównym serwerem.

### 3.4.3 Rekord NS

Informacja dla serwera DNS o adresach pozostałych serwerów. Rekordy te mają wskazywać na rekordy typu A, które należy utworzyć w pliku.

### 3.4.4 Rekord A

Rekord używany do mapowania nazw, na adresy.

---

### 3.4.5 Rekord CNAME

Rekord ten jest rozszerzeniem rekordu A, czyli przekierowuje “nazwę2 na nazwę1”, gdzie nazwa1 jest wcześniej nakierowana np. na adres 192.168.0.1.

### 3.4.6 Rekord MX

Rekord *Mail Exchange* powstały na potrzeby usługi poczty elektronicznej. Przy pomocy tego rekordu oznacza się serwery poczty. Tak jak rekord NS, rekord MX musi być nakierowany na nazwę, która jest rozwiązana rekordem typu A.

### 3.4.7 Rekord PTR

Rekord mapujący adres IP na nazwę hosta. Używa się go w zapytaniach typu *reverse DNS*

---

### 3.5 Test przy pomocy narzędzia dig.

```
root@debian:~# dig test.pl

;<>> DiG 9.16.22-Debian <>> test.pl
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 22862
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: b8c0bb5c3abf2da401000000621fc6c8b768906396f3f289 (good)
;; QUESTION SECTION:
;test.pl.                                IN      A

;; AUTHORITY SECTION:
test.pl.                                600      IN      SOA     dns1.test.pl. dane_admina.gmail.com. 2007010401 3600
600 86400 600

;; Query time: 0 msec
;; SERVER: 192.168.0.1#53(192.168.0.1)
;; WHEN: Wed Mar 02 20:34:32 CET 2022
;; MSG SIZE rcvd: 126
```

Rysunek 7: Wynik komendy Dig dla domenty test.pl

```
root@debian:~# dig dns1.test.pl

;<>> DiG 9.16.22-Debian <>> dns1.test.pl
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 15664
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: b82130bdf215ba7401000000621fc70ecb32403fd5820474 (good)
;; QUESTION SECTION:
;dns1.test.pl.                            IN      A

;; ANSWER SECTION:
dns1.test.pl.                            3600     IN      A       192.168.0.1

;; Query time: 0 msec
;; SERVER: 192.168.0.1#53(192.168.0.1)
;; WHEN: Wed Mar 02 20:35:42 CET 2022
;; MSG SIZE rcvd: 85

root@debian:~#
```

Rysunek 8: Wynik komendy Dig dla dns1.test.pl

Jak widać domena zwraca odpowiednie wartości rekordów, jeżeli zostanie “wypyтана” przez narzędzie *dig*.