

DNS priručnik

Dinko Korunić <dinkoDOTkorunicATinfomarDOThr>

verzija 1.5

Licenca:

- [Imenovanje-Nekomercijalno-Bez prerada 3.0 SAD](#)

Slobodno smijete:

- umnožavati, distribuirati i javnosti priopćavati djelo

Pod sljedećim uvjetima:

- **Imenovanje.** Morate priznati i označiti autorstvo djela na način kako je specificirao autor ili davatelj licence (ali ne način koji bi sugerirao da Vi ili Vaše korištenje njihova djela ima njihovu izravnu podršku).
- **Nekomercijalno.** Ovo djelo ne smijete koristiti u komercijalne svrhe.
- **Bez prerada.** Ne smijete mijenjati, preoblikovati ili prerađivati ovo djelo.
- U slučaju daljnjeg korištenja ili distribuiranja morate drugima jasno dati do znanja licencne uvjete ovog djela.
- Od svakog od tih uvjeta moguće je odstupiti, ako dobijete dopuštenje nositelja autorskog prava.
- Ništa u ovoj licenci ne narušava ili ograničava autorova moralna prava.

Prethodno ni na koji način ne utječe na zakonska ograničenja autorskog prava.

Zagreb, srpanj 2008.

Sadržaj

1. Uvod	4
1.1. Domensko ime	4
1.2. Domene	5
1.3. Domenski registri	6
1.4. DNS rezolucija	8
1.5. DNS međuspremnici	10
1.6. Reverzna rezolucija	11
1.7. DNS protokol i komunikacija	12
1.8. DNS klase i zapisi	15
1.9. Primarni i sekundarni NS, prijenos zone	17
1.10. Prijenos zone i poboljšanja	19
1.11. Delegacija	21
1.12. DNS dodaci i neki detalji	23
1.13. DNS sigurnost	24
2. DNS alati	27
2.1. Naredba host	27
2.2. Naredba dig	28
2.3. Naredba dnswalk	30
2.4. Naredba fpdns	31
2.5. Naredba nslint	32
2.6. Naredba zonecheck	32
2.7. Naredba dnstop	33
3. Bind9 poslužitelj	35
3.1. Konfiguracija općenito	35
3.2. Komentari	36
3.3. Parametri rada servisa	36
3.4. Pristupne liste	39
3.5. Odjeljak za zapisnike	40
3.6. Odjeljak kontrole	41
3.7. Odjeljak ključeva	42
3.8. Server odjeljak	42
3.9. Odjeljak konfiguracije pogleda	43
3.10. Umetnuta konfiguracijska datoteka	44
3.11. Odjeljak za zone	45
3.12. Konfiguracija zona	47
4. Djbdns poslužitelj	52
4.1. Dnscache	53
4.2. Tinydns	55
4.3. Tinydns zone	57
4.4. Axfrdns	62
4.5. Pravila za tcpserver	63
4.6. Walldns	65
4.7. Rbldns	65
5. MaraDNS	66
5.1. CSV1 format zone	67
5.2. CSV2 format zone	70
5.3. Mararc konfiguracijska datoteka	77
6. PowerDNS	78

7. Unbound	79
8. Dnsmasq	80
9. NSD	81
10. Primjeri konfiguracija	82
10.1. Bind9 konfiguracija - named.conf	82
10.2. Bind9 forward zona - hosts_fsb.db	84
10.3. Bind9 reverse zona - db.127	84
10.4. Bind9 wildcard zona - blockeddomain.hosts	85
10.5. Bind9 prazna zona - db.empty	85
10.6. Bind9 reverse zona - hosts_116.rev	85
10.7. Bind9 MS Active Directory kompatibilna zona	86
10.8. TinyDNS zona	87
10.9. MaraDNS CSV1 zona	87
10.10. MaraDNS CSV2 zona	87
11. Literatura	89

Primjer 1: Domenska imena, FQDN, labele	5
Primjer 2: TLD-ovi	6
Primjer 3: TLD, SLD i 3LD	6
Primjer 4: Međunarodni TLD-ovi	6
Primjer 5: Lista vršnih ICANN DNS poslužitelja	7
Primjer 6: Rekurzivni DNS upit	9
Primjer 7: Standardne i reverzne adrese	12
Primjer 8: Različiti RR-ovi u svijetu i kod nas	15
Primjer 9: SOA polja u praksi	20
Primjer 10: Reverzna delegacija bez klasa	22
Primjer 11: Kružno posluživanje	23
Primjer 12: Korištenje naredbe host	27
Primjer 13: Korištenje naredbe dig	29
Primjer 14: Korištenje naredbe dnswalk	30
Primjer 15: Korištenje naredbe fpdns	31
Primjer 16: Korištenje naredbe nslint	32
Primjer 17: Korištenje naredbe zonecheck	32
Primjer 18: Komentari u named.conf datoteci	36
Primjer 19: Options odjeljak iz named.conf datoteke	38
Primjer 20: Definiranje pristupnih listi u named.conf	39
Primjer 21: Korištenje logging direktive	41
Primjer 22: Controls odjeljak iz named.conf datoteke	42
Primjer 23: Ključ za rndc program i za Bind servis	42
Primjer 24: Korištenje server odjeljka	43
Primjer 25: Razdijeljeni DNS kroz view direktive	44
Primjer 26: Umetnute konfiguracijske datoteke	44
Primjer 27: Korištenje parametara unutar zonskih datoteka	48
Primjer 28: Kratice za dnscache	54
Primjer 29: Područje djelovanja u tinydns zoni	62
Primjer 30: Zamjenski zapisi u tinydns zoni	62
Primjer 31: Tcpserver pravila za axfrdns	64
Primjer 32: CSV1 konfiguracija za MaraDNS	67
Primjer 33: CSV2 konfiguracija za MaraDNS	71

1. Uvod

Na današnjem Internetu je tzv. DNS jedan od osnovnih servisa, te se praktički podrazumijeva njegovo shvaćanje i ispravna uporaba - čije ćemo temelje postaviti u ovoj kratkoj kuharici. Kako svaki priručnik počinje sa teoretskim uvodom, tako će i ovo uvodno poglavlje sadržavati neke osnovne pojmove nužne za razumijevanje i kasniju praktičnu primjenu u idućim poglavljima.

DNS (Domain Name System) je strogo hijerarhijski distribuirani sustav u kojem se mogu nalaziti različite informacije, prvenstveno one o IP adresama i slovni nazivima za računala. **Slovni naziv računala** (engl. hostname) je jedinstveno simboličko ime unutar pojedine mreže kojim se koriste neki protokoli (SMTP, NNTP) za elektroničku identifikaciju nekog računala. Takvi slovni nazivi mogu biti samo jedna riječ, ako se recimo radi o lokalnoj mreži; ili nekoliko riječi odvojenih točkama. U potonjem slučaju riječ je o **domenskom imenu** (engl. domain name) o kojem ćemo detaljnije nešto kasnije. Klijentima DNS informacije pružaju DNS **poslužitelji** koristeći DNS protokol za komunikaciju kako sa klijentima tako i međusobno.

Svrha DNS sustava je pojednostavljivanje komunikacije među računalima u smislu lakšeg pamćenja slovni naziva kao i mogućnosti tematskih i inih grupiranja računala koja nisu nužno fizički blizu (fizički blizu u smislu slijednih IP adresa). Očito je bitno ugodnije u svakodnevnom radu koristiti i pamtiti slovna, simbolička imena umjesto odgovarajućih IP adresa.

Sam DNS sustav je naravno puno širi, te obuhvaća tri osnovne funkcije sa različitim segmentima koje ćemo definirati u daljnjem tekstu:

1. DNS imenički prostor, problematiku imenovanja i pravila: karakteristike su hijerarhijska struktura, imenička struktura i pravila imenovanja te specifikacije domena,
2. registraciju domena i ine administrativne probleme: hijerarhijsku strukturu nadležnih tijela, hijerarhiju vršnih nadležnih tijela (TLD), procedure registracije sekundarnih domena, administraciju DNS zona i administraciju hijerarhije,
3. poslužitelje i proces rezolucije: DNS zapisi i zone, tipovi DNS poslužitelja sa različitim ulogama, procesi rezolucije, DNS poruke, formati i zapisi.

1.1.Domensko ime

Domensko ime je simboličko ime računala na Internetu koje ga uglavnom (postoji mogućnost da više računala dijeli jedno domensko ime) jedinstveno označuje. DNS sustav vrši preslikavanje domenskog imena u jednu ili više IP adresa te obrnuto, preslikavanje jedne ili više IP adrese u jedno domensko ime. Na većini modernih operacijskih sustava se DNS sustav koristi implicitno, pa je moguće nekom računalu na Internetu pristupiti kako kroz odgovarajuću IP adresu, tako i kroz domensko ime - ako ono postoji.

Domensko ime se često naziva i **labela** (engl. label). Po strogoj definiciji labela je alfanumerički niz znakova sa maksimalno 63 znaka. Jedini dakle ispravni i dozvoljeni znakovi u pojedinoj DNS labeli su:

- Slova od A do Z, odnosno od a do z,
- Brojevi od 0 do 9,
- Znak "-".

Očigledan je manjak podrške za lokalizaciju domena, a nažalost ista problematika nije ni dan danas u potpunosti riješena odgovarajućim općeprihvaćenim standardom.

Više takvih labela se međusobno odvaja točkama, a sve one zajedno tvore **domensko ime**, koje se u takvoj potpunoj formi (navedene su sve labela) zove i **FQDN** (Fully Qualified Domain Name). Takvo ime je ukupne maksimalne dužine od 255 znakova, a različito je od običnog domenskog imena (koje može biti i kratkog oblika, sadržavajući svega dio labela) po tome što predstavlja apsolutnu stazu unutar DNS hijerarhije.

Primjer 1: Domenska imena, FQDN, labela

FQDN:	www.srce.hr., jagor.srce.hr
labela:	www, jagor, srce, hr
ime računala:	www, regoc, jagor, kosjenka
domensko ime:	jagor.srce, www

Napomenimo još jednom - svaka labela se sastoji od isključivo alfanumeričkih znakova i znaka "-" (dakle ASCII znakovi od A do Z i znak "-"), pri čemu se labela ne razlikuju po velikim i malim slovima. Danas je u procesu prihvatanja novi sustav koji bi trebao dozvoliti i ne-ASCII znakove u labelama, tzv. IDNA (engl. Internationalizing Domain Names in Applications) baziran na Punycode enkodiranju Unicode nizova. Da bi se FQDN dodatno razlikovao od labela odnosno standardnih (ne nužno potpunih) domenskih imena, česta je konvencija dodavanja **dodatne točke** (znaka ".") na kraj domenskog imena.

Da ponovimo: domensko ime se sastoji od dvije ili više labela odvojenih točkama. Krajnje desna labela se naziva TLD (više o TLD-ovima u nastavku), a svaka druga labela lijevo je **poddomena** - domena koja je hijerarhijski ispod prethodne. Ukupno maksimalno podjela može biti 127, dok se držimo zadane granice od 255 znakova za FQDN. Na kraju, labela koja je krajnje lijeva je kratko ime računala (već spomenuti slovni naziv računala, dakle bez domene).

1.2.Domene

U pojedinoj domeni, odnosno **domenskom prostoru** ne mogu postojati dvije iste labela - što znači niti dvije poddomene niti dva računala. Domenska imena su obično grupirana; ona završavaju pojedinom grupom labela za koje postoje točno definirana pravila. Takve završne labela se nazivaju **TLD** (engl. Top-Level Domain) imena, kojih postoje dva tipa:

- Geografski bazirane domene, tzv. **ccTLD** (engl. country code TLD) domene koje predstavljaju državni dvoznakovni kod temeljen oko ISO-3166 standarda, a danas ih ima preko 243 u upotrebi,
- Generičke domene, tzv. **gTLD** (engl. generic TLD) domene koje se obično sastoje od 3 ili više znakova.

Primjer 2: TLD-ovi

gTLD: .com, .net, .org, .biz, .info, .name, .museum, .travel, .xxx
ccTLD: .us, .fr, .es, .de, .it, .jp, .ie, .co.uk, ...
ccTLD izvan ISO 3166-1: .ac, .su, .tp, .uk, .yu, .eu

Za dodjelu i upravljanje problematikom domena, zaduženo je **ICANN** (Internet Corporation for Assigned Names and Numbers) neprofitno tijelo. Ova relativno mlada organizacija je preuzela poslove koje je nekad obavljala **IANA** (Internet Assigned Numbers Authority). Specifično, riječ je o upravljanju dodjeljivanjem domena i IP adresa, pri čemu se lokalna registracija IP adresa u predaje pojedinačnim RIR-ovima (Regional Internet Registries). Svaki RIR alocira adrese za različiti dio svijeta. Osim TLD-ova, u DNS terminologiji se još koriste i **SLD** (engl. Second-Level Domain) odnosno domene drugog stupnja te **3LD** (engl. Third-Level Domain), domene trećeg stupnja. Ukupno u svijetu trenutno postoji 258 TLD-ova (zajedno ccTLD i gTLD).

Primjer 3: TLD, SLD i 3LD

TLD: .hr
SLD: .fer.hr
3LD: .esa.fer.hr

1.3.Domenski registri

Slično kao i za IP adrese, postoje **domenski registri**, baze podataka o domenama i odgovarajućim IP adresama, po jedan za svaku TLD. Oni kao uslugu daju domenska imena za vlastitu TLD te omogućavaju ostatku svijeta pregled informacija o registracijama pojedinih domena. Domenski registri se inače nazivaju **NIC** (Network Information Centre), te su najčešće neprofitne ili državne organizacije. Informacije o registracijama su dostupne kroz **Whois** sustav, pa je tako za Europu nadležan whois.ripe.net poslužitelj (primjer dobrog Whois klijenta je **Jwhois**, sa ugrađenim bazama lokalnih registara). Same registracije se najčešće dešavaju na principu "tko prvi, njemu djevojka", iako pojedini mogu formirati složene politike zbog zaštićenih imena, itd. Svaki registar upravlja DNS poslužiteljima za specifični TLD, pa je to za Hrvatsku (.HR) dns.srce.hr kojim upravlja HR-DNS služba za CARNet, sa 28 tisuća registriranih domena u 2004. Dakle, za ccTLD-ove su obično nadležne vlade pojedine države, dok je za gTLD nadležan isključivo ICANN.

Primjer 4: Međunarodni TLD-ovi

TLD	opis	URL
AC	ccTLD - Ascension Island	Network Information Center (AC Domain Registry) (http://www.nic.ac/)
AD	ccTLD - Andorra	Servei de Telecomunicacions d'Andorra (http://www.nic.ad)
AE	ccTLD - United Arab Emirates	Emirates Telecommunications Corporation (http://www.nic.ae)
AERO	gTLD - AERO (Aviation Community)	Societe Internationale de Telecommunications Aeronautique S.C. (http://www.information.aero)
...
HR	ccTLD - Croatia/Hrvatska	CARNet - Croatian Academic and Research Network (http://www.dns.hr)
...

Naravno, za osnovnu domenu je također nadležan ICANN, koji regulira upravljanjem 13 **vršnih DNS poslužitelja** (engl. root servers). Rečeni DNS poslužitelji su uglavnom fizički smješteni u USA. Dio adresa vršnih poslužitelja se distribuira anycast tehnologijom koristeći BGP protokol kako bi se omogućila decentralizacija - na taj način se veliki broj distribuiranih čvorova u svijetu pojavljuje kao jedinstveni čvor odnosno servis. Trenutno je na 13 vršnih poslužitelja raspoređeno ukupno stotinjak (svibnja 2007. bilo je 117 vršnih poslužitelja) fizičkih poslužitelja diljem svijeta. Uzgred, zanimljivo je spomenuti da se smatra kako je prosječno 98% prometa (a prosječan čvor prima oko 2000 DNS upita u sekundi) koje vršni DNS poslužitelji primaju zapravo nepotreban promet, rezultat različitih konfiguracijskih i inih grešaka.

Primjer 5: Lista vršnih ICANN DNS poslužitelja

ime	operator	lokacija	IP adresa	ASN
A	VeriSign Naming and Directory Services	Dulles, Virginia, USA	IPv4: 198.41.0.4	19836
B	Information Sciences Institute	Marina Del Rey, California, USA	IPv4: 192.228.79.201 IPv6: 2001:478:65::53	tba
C	Cogent Communications	anycast	IPv4: 192.33.4.12	2149
D	University of Maryland	College Park, Maryland, USA	IPv4: 128.8.10.90	27
E	NASA Ames Research Center	Mountain View, California, USA	IPv4: 192.203.230.10	297
F	Internet Systems Consortium, Inc.	anycast	IPv4: 192.5.5.241 IPv6: 2001:500::1035	3557
G	U.S. DOD Network Information Center	Columbus, Ohio, USA	IPv4: 192.112.36.4	568
H	U.S. Army Research Lab	Aberdeen Proving Ground, Maryland, USA	IPv4: 128.63.2.53 IPv6: 2001:500:1::803f:235	13

ime	operator	lokacija	IP adresa	ASN
I	Autonomica/NORDU net	anycast	IPv4: 192.36.148.17	29216
J	VeriSign Naming and Directory Services	anycast	IPv4: 192.58.128.30	26415
K	Reseaux IP Europeens - Network Coordination Centre	anycast	IPv4: 193.0.14.129 IPv6: 2001:7fd::1	25152
L	Internet Corporation for Assigned Names and Numbers	Los Angeles, California, USA	IPv4: 198.32.64.12	20144
M	WIDE Project	anycast	IPv4: 202.12.27.33 IPv6: 2001:dc3::35	7500

Postoje i različite organizacije koje nude alternativne vršne DNS poslužitelje, nudeći najčešće i vlastiti skup TLD-jeva, nekompatibilan sa ICANN-ovom listom. Neki primjeri su ORSC (Open Root Server Confederation), OpenNIC, Pacific Root, New.Net; no najorganiziraniji je **ORSN** (Open Root Server Network) koji ima direktnu kompatibilnost sa ICANN-ovom bazom - što u praksi znači dodatnu redundanciju posebice pogodnu za Europske TLD-jeve.

Jedan od glavnih problema sa čvornim poslužiteljima je njihova nejednolika geografska distribucija. Naime, lokacija poslužitelja direktno odgovara sa financijskom situacijom regije što je očiti posljedak cijena pristupa ostatku Interneta, održavanja čvornog poslužitelja i potrebne infrastrukture. Ovo pak uzrokuje da siromašne zemlje nemaju lokalni čvorni poslužitelj (npr. Sri Lanka i Pakistan), čime se bitno povećava latencija i smanjuje pouzdanost pristupa Internet resursima i unutar vlastite zemlje. Čak i ccTLD poslužitelji često nisu unutar vlastite zemlje: tek ih je dvije trećine unutar vlastitih zemalja, a postoji i dobar dio koji je izvan centralnog i dobro povezanog dijela Interneta. Ispadom takvog ccTLD, sve za što je on bio zadužen postaje nedostupno ma gdje god to bilo (geografski ili po Internet spojenosti).

1.4.DNS rezolucija

Svaki se funkcionalni DNS sustav nužno sastoji se od tri dijela:

- DNS **klijent** (engl. resolver), program koji se izvršava na klijentskom računalu i koji formira određeni DNS zahtjev. Takav program ne mora biti nužno samostojeći servis, on je na većini Unixoida najčešće ugrađen u standardnoj biblioteci u formi sistemskih poziva koje pozivaju različiti korisnički programi,
- **Rekurzivni** (engl. recursive) DNS **poslužitelj**, koji nakon dobivenih upita za klijenta obavlja pretraživanje kroz DNS stablo i vraća nazad odgovore klijentima,

- **Autoritativni** (engl. authoritative) DNS **poslužitelj**, koji odgovara na upite rekurzivnih poslužitelja te vraća ili završni odgovor ili zbog delegiranja vraća referencu na neki drugi autoritativni DNS poslužitelj.

Sam proces primanja zahtjeva i njihove obrade te vraćanja odgovora se naziva DNS **rezolucija** (engl. name resolution). Pojednostavljeno, osnovna rezolucija je proces pretvorbe domenskog imena u IP adresu: prvo tražimo autoritativni DNS poslužitelj, a zatim mu šaljemo upit za adresom, na koji on odgovara sa traženom adresom. Budući da je DNS strogo distribuirana baza, ona je raspodijeljena po mnogo različitih poslužitelja. No, očigledno je da zbog raspodijeljenosti rezolucija obično ne može biti obavljena kroz samo jedan upit i odgovor, već najčešće zahtijeva dužu komunikaciju i niz upita i odgovora. Najčešća je situacija da klijent šalje zahtjeve **lokalnom** DNS **poslužitelju** (nadležan za klijentsko računalo, obično dodijeljen od ISP-a ili ustanove u kojoj se nalazi klijentsko računalo), koji predstavlja rekurzivni poslužitelj i obavlja upite te zatim vraća odgovor klijentu. Dakle, najveći i najkompliciraniji dio procedure predstavlja traženje autoritativnog poslužitelja u složenoj DNS hijerarhiji.

Što se samih tipova DNS rezolucije tiče, postoje dva osnovna tipa prolaska kroz DNS hijerarhiju da bi se otkrio točan zapis. Oni se razlikuju po tome tko obavlja većinu posla oko saznavanja podataka i njihove obrade, a prvenstveno se pojavljuju kad obrada određenog DNS upita zahtijeva nekoliko koraka (dakle, lokalni DNS poslužitelj nema sve informacije):

- **Iterativni** - kada klijent šalje dotične upite, poslužitelj mora odgovoriti jednim od dva moguća odgovora: a) odgovorom na zahtjev ili b) imenom drugog DNS poslužitelja (vrši se **delegiranje**) koji ima više podataka o traženom upitu. U ovakvom tipu upita najveći dio posla obavlja klijent iterirajući akcije upit-odgovor i prolazeći kroz DNS hijerarhiju.
- **Rekurzivni** - kada klijent šalje rekurzivni upit, poslužitelj preuzima posao pronalaženja informacija o traženom upitu. Dakle, ono što je u iterativnom obavljao klijent, kod rekurzivnih upita obavlja poslužitelj - obrađuje informacije i šalje nove upite drugim poslužiteljima sve dok ne pronađe traženo. Dakle, klijent šalje svega jedan zahtjev te dobiva ili točnu informaciju koju je tražio ili poruku o grešci.

Očigledno je rekurzivan način pretraživanja vrlo povoljan za klijente, ali može znatno opteretiti DNS poslužitelje (na stranu i potencijalni problem trovanja DNS poslužitelja o kojem će kasnije biti riječi), pa se takve forme upita obično eksplicitno dozvoljavaju samo računalima iz lokalne mreže, dakle računalima kojima je dotični DNS poslužitelj nadležan. I isključivo njima.

Primjer 6: Rekurzivni DNS upit

Kao primjer, pokazat ćemo razrješavanje rekurzivnog upita u potrazi za "www.carnet.hr":

1. lokalni DNS poslužitelj dobiva rekurzivni zahtjev,
2. pretražuje lokalnu listu vršnih DNS poslužitelja,

3. rekurzivni proces započinje iterativnim upitom jednom (slučajni odabir) od vršnih DNS poslužitelja za `www.carnet.hr` adresom,
4. odabrani vršni DNS poslužitelj odgovara na upit sa delegacijom na `ccTLD` poslužitelj za `hr` domenu,
5. lokalni DNS poslužitelj zatim šalje iterativni upit `hr` DNS poslužitelju (primjerice `dns.srce.hr` odnosno `161.53.3.7`) za `www.carnet.hr` adresom,
6. `hr` DNS poslužitelj odgovara sa delegacijom na `carnet.hr` DNS poslužitelj (`dns.carnet.hr`, odnosno `161.53.123.3`)
7. lokalni DNS poslužitelj šalje iterativni upit `carnet.hr` DNS poslužitelju (`dns.carnet.hr`) za `www.carnet.hr` adresom
8. `carnet.hr` DNS poslužitelj (`dns.carnet.hr`) odgovara sa autoritativnim odgovorom, odnosno IP adresom za `www.carnet.hr` (primjerice `161.53.160.25`)
9. lokalni DNS poslužitelj odgovara klijentu sa odgovorom dobivenim od autoritativnog poslužitelja (u našem slučaju `161.53.160.25`).
10. ovime proces završava.

Već smo spomenuli da je DNS vrlo strogo hijerarhijski baziran - praktički svaka pretraga za nekom DNS informacijom počinje od čvornog DNS računala, od vrha DNS **stabla**. Prolazak kroz DNS stablo je silazak po granama stabla, gdje je svaki čvor jedan DNS poslužitelj, nadležan za svoj dio DNS prostora. Osnovni preduvjet pronalaženja čvora stabla je lokalna lista 13 vršnih DNS poslužitelja, koji dalje delegiraju pretragu po zapisima. DNS stablo je dakle hijerarhijski složen skup DNS poslužitelja, gdje svaka domena i poddomena ima jednog ili više autoritativnih DNS poslužitelja. Dotični poslužitelji (čvorovi stabla) su nadležni (ili mogu delegirati dalje) za "sve" domene ispod njih, servirajući podatke drugima na upit. Hijerarhijski raspored poslužitelja upravo mora odgovarati rasporedu domena i odgovarajućeg domenskog prostora.

U svakodnevnoj upotrebi, osim domena i labela pojavljuje se i pojam **zona**. Zona kao takva predstavlja dio ukupnog domenskog prostora, te se prostire od jedne točke - jednog DNS poslužitelja zaduženog za tu zonu, odnosno autoritativnog za tu zonu - dalje do krajnjih čvorova ili do početaka neke druge zone. Tehnički zona je dakle dio domene, iako se može prostirati i na cijelu domenu.

1.5.DNS međuspremnici

DNS je sustav sa ovim osnovnim načinima pretraživanja (iterativni i rekurzivni silazak kroz DNS stablo) vrlo neefikasan, budući da svaki upit implicitno znači novi prolazak po stablu, počevši od vršnih DNS poslužitelja. Jasno, kada bi se u stvarnom svijetu nužno svaki put prolazilo od početka DNS stabla do kraja, do traženog zapisa - proces DNS rezolucije bi trajao i trajao, a opterećenje DNS poslužitelja bi postalo pretjerano veliko, sve veće i veće sa porastom

broja računala na Internetu. No, eskalaciju ovog problema prilično je smanjio jednostavan princip spremanja kako pozitivnih (uspješnih) tako i negativnih (neuspješnih) rezultata DNS upita na DNS poslužiteljima. Naime, formiranje međuspremnik (engl. cache) DNS rezultata je odgovor na dva jednostavna fenomena prisutnim u računalnim mrežama i računalima općenito:

- Veće su šanse da se će pristupati nekom resursu ako se nedavno pristupalo nekom drugom (prostorno) bliskom resursu - što je tzv. **prostorna lokalnost reference**,
- Ako se samom tom resursu nedavno pristupalo, to su veće šanse da će mu se ponovno pristupati - što je tzv. **vremenska lokalnost reference**.

Praksa pokazuje da se vrlo često šalju slični ili isti DNS upiti u vremenski bliskim periodima. Stoga svi moderni DNS poslužitelji imaju interne međuspremnik o nedavnim DNS upitima koji im omogućavaju da pribave odgovor ili dio odgovora iz međuspremnik (obično u privremenoj memoriji). DNS spremnik se nalaze i na većini DNS klijenata, obavljajući isti posao kao i na DNS poslužiteljima. Na taj način se spremaju rezultati već obavljenih upita i na klijentskim računalima, smanjujući na taj način promet prema poslužiteljima i njihovo opterećenje. Tako spremljeni odgovori će biti "duže" spremljeni kod klijenata, budući da je svaki spremnik ograničene veličine i dovoljan broj upita "istiskuje" stare ili nekorištene odgovore (po FIFO, LRU, GDSF ili nekom drugom principu). Međuspremnik kao takvi više poboljšavaju performanse što su bliže klijentu, ali daju bolju pokrivenost što su dalje od klijenta. Podaci spremljeni u spremnicima imaju svoja vremena života, TTL (Time To Live), pa se time osigurava da zastarjeli podaci nužno nestaju iz spremnika.

Današnji moderni DNS poslužitelji će pri prihvatu DNS upita obaviti pretraživanje vlastitih spremnika kao i lokalne DNS baze, pokušavajući što više skratiti vremenski "skup" prolazak kroz DNS stablo. Nažalost, sekundarni efekt postojanja TTL vremena za DNS zapise je da utiče i na **vrijeme širenja podataka** (engl. propagation time) po DNS stablu, budući da se promjene između ne vide - sve do eksplicitnog nestajanja zapisa zbog TTL-a. Stoga je razumijevanje koncepta TTL-a apsolutna nužnost ne samo za ispravan rad pojedinog DNS poslužitelja već i globalno: primjerice za popularne adrese je poželjno koristiti visoke TTL vrijednosti zbog smanjenja opterećenja na cijeli DNS sustav.

1.6.Reverzna rezolucija

Do sada smo samo spominjali standardnu unaprijednu (engl. forward) rezoluciju kod koje se DNS imena pretvaraju u IP adrese. U standardnoj komunikaciji na Internetu nužno je moći vršiti rezoluciju u oba smjera, što će reći i u unazadnom obliku (engl. reverse) - primjerice za provjeru spada li određena adresa u kakvu domenu i sl. No, problem kod **reverzne DNS rezolucije** je da se poslužitelji razlikuju prvenstveno po labelama odnosno po domenama za koje su nadležni, a ovdje imamo samo IP adresu kao početnu informaciju.

Kako bi se pojednostavio (nekad se koristio neefikasni inverzni upit IQUERY koji danas uglavnom više nije u upotrebi) i uopće omogućio ovaj proces, formirana je dodatna hijerarhija u vidu **IN-ADDR.ARPA** domene. Riječ je o domenskom prostoru koji se sastoji od četiri nivoa poddomena, a svaki nivo odgovara jednom dijelu IP adrese. Reverznom DNS rezolucijom odnosno prolaskom kroz dotična četiri nivoa, dolazi se do čvora za traženu IP adresu koji pokazuje na odgovarajuće domensko ime.

Svaki nivo unutar in-addr.arpa domene se sastoji od 256 poddomena (0, 1, ..., 255). **Reverzna DNS adresa** se formira od čvorova unutar reverzne domene, a identična je obrnuto zapisanoj IP adresi sa in-addr.arpa sufiksom, pa upiti nad takvom adresom daju kao povratnu informaciju "standardnu" DNS adresu:

Primjer 7: Standardne i reverzne adrese

fly.srk.fer.hr	A	161.53.74.66
66.74.53.161.in-addr.arpa	PTR	fly.srk.fer.hr

1.7.DNS protokol i komunikacija

DNS poslužitelj koristi standardne portove dodijeljene od IANA-e: **TCP/53** i **UDP /53**. Na njima osluškuje zahtjeve, te može bilo sa dotičnih bilo sa nekog visokog porta (port veći od 1024, ovisno o konfiguraciji poslužitelja) poslati odgovor u vidu traženih zapisa odnosno **RR**-ova (engl. resource record). Standardno se uvijek koristi UDP za upite, a komunikacija se uglavnom svodi na jedan UDP upit i jedan UDP odgovor. TCP komunikacije se koristi jedino kad veličina odgovora prelazi 512 bajtova ili za grupne prijenose DNS informacija, tzv. **prijenos zone** (engl. zone transfer).

Standardni DNS upit je obično vrlo jednostavan, sadrži uglavnom samo adresu koja se želi razriješiti - no odgovori su vrlo komplicirani budući da sadrže sve adrese i zamjenske adrese koje su rezultat upita. Stoga se odgovori obično sažimaju posebnim algoritmima, eliminirajući nepotrebne podatke i smanjujući samu veličinu UDP datagrama. U slučaju da i dalje veličina paketa prelazi 512 bajtova, šalje se parcijalna poruka u obliku UDP paketa sa posebnim bitom postavljenim (TC=1), koji označuje da se upit mora ponoviti koristeći TCP. Rečena maksimalna veličina paketa je ujedno i odgovor zašto postoji svega 13 vršnih DNS poslužitelja: upravo se lista od 13 IP adresa odgovarajuće sprema u jedan paket od 512 bajtova.

Za DNS upite i odgovore se koristi tzv. opći oblik poruke, koji se sastoji od 5 odjeljaka. Dotični se popunjavaju kako upitom od klijenta, tako i odgovorom od poslužitelja i u oba slučaja i podacima u zaglavlju koji su nužni da se proces obavi ispravno i uspješno. Dotični odjeljci sa sadržajem su:

- **Zaglavlje** (engl. header) - nužna polja koja definiraju tip poruke i pružaju klijentu ili poslužitelju važne informacije o poruci. Također, u zaglavlju se nalaze i brojači zapisa u drugim odjeljcima poruke. Zaglavlje je prisutno u svim porukama i fiksne je veličine od 12 bajtova.

Jedna od važnijih zastavica u zaglavlju je i QR koja označava da li je poruka upit ili odgovor,

- **Pitanje** (engl. question) - jedan ili više upita klijenta prema DNS poslužitelju,
- **Odgovor** (engl. answer) - jedan ili više RR-ova koji su odgovor na klijentov upit,
- **Autoritet** (engl. authority) - jedan ili više RR-ova koji predstavljaju delegaciju na autoritativne poslužitelje, odnosno pokazuju na autoritativne DNS poslužitelje koji se mogu koristiti za nastavak DNS rezolucije,
- **Dodatno** (engl. additional) - jedan ili više RR-ova koji sadrže različite dodatne informacije vezane uz upit, ali dotične nisu nužne za potpunost odgovora ili upita; primjerice IP adresa DNS poslužitelja spomenutog u polju za autoritet.

Moguće zastavice u zaglavlju DNS poruke su sljedeće:

- **ID** (engl. identifier) - riječ je o 16bitnom identifikacijskom broju koje stvara računalo ili uređaj koji šalje DNS upit. Poslužitelj u poruci mora odgovoriti sa istim takvim brojem, što omogućava klijentu da prepozna par upit-odgovor,
- **QR** (engl. query/response flag) - služi za razlikovanje upita i odgovora. Postavljena je na 0 za upit od klijenta, a 1 za odgovor od poslužitelja,
- **Opcode** - označava tip operacije koji se nalazi u poruci: 0 je standardni upit (QUERY), 1 je zastarjeli tip inverznog upita (IQUERY) koji se više ne koristi, 2 je upit za statusom poslužitelja (STATUS), 3 se ne koristi, 4 je posebna poruka upozorenja koja se koristi u grupnom prijenosu DNS zapisa (NOTIFY), 5 je poruka za osvježanje DNS zapisa koja se koristi u dinamičkom DNS-u (UPDATE),
- **AA** (engl. authoritative answer flag) - zastavica će biti postavljena na 1 ako je poslužitelj koji šalje odgovor autoritativan za zonu koja je dana u odjeljku pitanja, a u suprotnom će biti 0,
- **TC** (engl. truncation flag) - zastavica koja kad je postavljena na 1 označava da je poruka nepotpuna budući da je bi ukupna veličina UDP poruke bila veća od 512 bajtova. Klijent tada može poslati novi zahtjev da bi dobio potpun odgovor, pa se najčešće ostvaruje novi zahtjev-odgovor koristeći TCP,
- **RD** (engl. recursion desired) - kada je dotična zastavica postavljena, označava da bi bilo poželjno da poslužitelj obavi rekurzivnu rezoluciju, ako to poslužitelj podržava. Odgovor koji poslužitelj šalje će zadržati isto stanje zastavice kao i u upitu,
- **RA** (engl. recursion available) - kada je postavljena zastavica, znači da poslužitelj koji šalje odgovor podržava rekurzivne upite, što klijenti najčešće "zapamte" za buduću komunikaciju sa dotičnim poslužiteljem,
- **Z** (engl. zero) - bitovi koji su uvijek postavljeni na 0,
- **Rcode** (engl. response code) - zastavica koja je u upitima uvijek na 0, ali u odgovorima indicira na tip greške koji se desio, odnosno da li je uspješno došlo do odgovora: 0 ukazuje da nije došlo do greške (engl. NoError), 1 znači da poslužitelj nije mogao odgovoriti zbog neispravnog oblika upita (engl. Format Error), 2 znači da je poslužitelj pretrpio

internu grešku u radu (engl. Server Failure), 3 indicira da objekt upita ne postoji u traženoj domeni - što može odgovoriti ili autoritativni poslužitelj ili lokalni poslužitelj iz negativnog međuspremnik (engl. Name Error), 4 ukazuje da poslužitelj ne podržava dotični tip upita (engl. Not Implemented), 5 znači da je poslužitelj odbio izvršiti upit, najčešće zbog pristupnih listi i konfiguracije (engl. Refused), 6 znači da domensko ime postoji kad ne bi trebalo (engl. YX Domain), 7 znači da RR postoji kad ne bi trebao (engl. YX RR Set), 8 znači da ne postoji RR koji bi trebao postojati (engl. NX RR Set), 9 ukazuje da DNS poslužitelj koji je primio zahtjev nije autoritativan za dotični prostor (engl. Not Auth), 10 ukazuje da zatraženi objekt u zoni/prostoru specificiranom u upitu (engl. Not Zone),

- **QDCOUNT** (engl. question count) je brojač upita u odjeljku pitanja poruke,
- **ANCOUNT** (engl. answer record count) je brojač RR-ova u odjeljku odgovora poruke,
- **NSCOUNT** (engl. authority record count) je brojač RR-ova u odjeljku autoriteta poruke,
- **ARCOUNT** (engl. additional record count) je brojač RR-ova u dodatnom odjeljku poruke.

I još ćemo pokazati kako izgleda oblik zaglavlja upita, koje definira sadržaj upita poslanog od klijenta prema poslužitelju. Ono se sastoji od nekoliko polja:

- **QName** (engl. question name) - sadrži objekt, domen u ili zonu koji su predmet upita,
- **QType** (engl. question type) - sadrži tip samog upita za upit koji dolazi od klijenta. Isti može sadržavati specifični broj koji odgovara tipu RR-a koji se traži ili pak neki od posebnih brojeva za posebne vrste upita: 251 odgovara zahtjevu za inkrementalni zonski prijenos (IXFR), 252 odgovara standardnom zahtjevu za prijenos zone (AXFR), 253 i 254 odgovaraju zastjermjelim upitima za zapise vezane uz e-mail (MAILA i MAILB upiti za MB, MG i MR zapisma), te 255 koji odgovara upitu za svim zapisima ("*").
- **QClass** (engl. question class) - označava koji se tip RR traži i može poprimiti vrijednost od 0 do 65535. Standardno je se koristi svega pet vrijednosti: 1 za Internet (IN) zapis, 3 za CHAOS, 4 za Hesiod (HS), 254 za prazni (NONE) tip i 255 za ANY upit. ANY klasa je zamjenski ("*"), dok se NONE obično koristi u dinamičkom DNS-u.

Poruka od DNS klijenta je primjerice sljedećeg oblika: klijent šalje UDP upit (QR=0, što označava upit, a ne odgovor) kao standardni upit (OPCODE=0) sa jednim zapisom u upitu (QDCOUNT=1). Upit uglavnom ne sadrži dodatne zapise niti u polju za odgovor, niti za autoritativni dio niti u polju za dodatne zapise (ANCOUNT=0, NSCOUNT=0, ARCOUNT=0). QNAME zapis označava primjerice domen u za kojom klijent pretražuje (QNAME = www.google.com.). Tip i klasa zapisa za kojom klijent pretražuje su QTYPE=1 (adresa računala) i QCLASS=1 (Internet adresa). Budući da veličina odgovora unutar 512 bajtova, TC=0.

Odgovor (QR=1) od poslužitelja na standardni upit (OPCODE=0) je primjerice sljedeći: poslužitelj je autoritativan za traženu domenu (AA=1), a podržava i rekurzivne upite (RA=1). Tijekom pretrage nisu utvrđene nikakve greške u upitu (RCODE=0) koji je sadržavao samo jedan zapis (QDCOUNT=1). Odgovor sadržava 3 RR-a (ANCOUNT=3) u polju odgovora, 6 zapisa u odjeljku za autoritet (ARCOUNT=6). Očigledno je da se originalni upit koristi za formiranje odgovora, pa se polje zaglavlja i polje pitanja kopiraju iz originalnog upita u odgovor, sa već navedenim promjenama.

1.8.DNS klase i zapisi

Kao što je već spomenuto, RR je jedan zapis, jedna jedinica u DNS sustavu. Svaki RR sadrži određene attribute, odgovarajuće za vlastiti tip; to mogu biti IP adresa, adresa za isporuku elektroničke pošte, niz teksta, DNS labela ili nešto treće. Svaki RR se sastoji od sljedećih komponenti, redom kojim se pojavljuju:

- Ime domene - uglavnom se koristi FQDN, a ako je zapisano kratko ime onda se automatski dodaje ime zone na kraj imena,
- TTL u sekundama, standardna vrijednost je minimalna vrijednost navedena u SOA zapisu (o ovome kasnije),
- **klasa zapisa** koji može biti Internet, Hesiod i Chaos,
- Tip zapisa: CNAME, PTR, A, MX, TXT, AAAA, A6, itd.
- Podaci za dotični tip zapisa - odgovaraju određenom tipu, ako sadržavaju ime domene koje nije FQDN, automatski se dodaje ime zone na kraj imena,
- Opcionalni komentar (dodan u ovisnosti o vrsti poslužiteljskog softvera).

Primjer 8: Različiti RR-ovi u svijetu i kod nas

esa.fer.hr.	22h30m57s	IN A	161.53.71.180
carnet.hr.	23h17m31s	IN NS	dns2.carnet.hr.
carnet.hr.	4h15m27s	IN MX	30 mx2.carnet.hr.
www.l.google.com.	5M	IN A	66.249.93.104
www.l.google.com.	5M	IN A	66.249.93.99
130.2.53.161.in-addr.arpa	86377	IN	PTR
jagor.srce.hr			
version.bind.	0S	CHAOS TXT	"9.2.2"

Klase zapisa (engl. resource record classes) su u osnovi povijesna ostavština, bez stvarne koristi danas. Budući da je DNS inicijalno vrlo generički оформljen, ideja je bila da će se kroz DNS nuditi imeničke usluge za više od jednog protokola (dakle, osim IP-a). Stoga svaki RR zapis ima i klasu, te općenito rečeno ona mora biti specificirana za svaki RR unutar lokalne zone. Danas se u praksi koristi jedino Internet klasa, pa se ona implicitno podrazumijeva kad u lokalnoj zoni nije eksplicitno navedena IN klasa.

Što se pak tiče tipova zapisa, postoji nekoliko osnovnih tipova:

- **A** (engl. address) - povezuje odgovarajuće domensko ime (labelu ili niz labela) sa IPv4 adresom (32bitna adresa). Danas je često moguće naći da više A zapisa pokazuje na istu IP adresu, što je sasvim legalno.

- **CNAME** (engl. canonical name) - omogućava da jedno domensko ime bude zamjensko ime za drugo. Takvo zamjensko ime dobiva sve osobine originala, uključujući i IP adrese i poddomene. No, ilegalno je u zoni imati ijedan zapis koji dijeli isto ime (labelu) kao i CNAME zapis; dakle, CNAME ne može koegzistirati niti s jednim drugim tipom za istu labelu, uključujući i praznu labelu. Također, niti jedan tip zapisa osim CNAME ne smije pokazivati na zamjensku adresu (odnosno na CNAME), budući da bi to omogućilo petlje i neispravne zapise u zoni.
- **MX** (engl. mail exchange) - označava koji su sve e-mail poslužitelji nadležni za dotičnu domenu. U slučaju da ovaj zapis ne postoji, e-mail se isporučuje koristeći A zapis dobiven rezolucijom iz odredišne domene. Osnovna funkcionalnost ovog mehanizma je pružiti mogućnost da postoji više e-mail poslužitelja za jednu domenu i da se definira točan redoslijed prema kojem ih se mora kontaktirati. Time se na jednostavan način omogućava usmjerivanje maila (engl. mail routing) kao i mogućnost raspodjele opterećenja između više poslužitelja. No, nažalost MX zapis ne omogućava e-mail servis na alternativnim portovima niti ne omogućava postavljanje težinskih vrijednosti za poslužitelje koji su istog prioriteta - kao što recimo SRV zapis omogućava. MX zapis funkcionira tako da klijent pri MX zahtjevu dobiva listu e-mail poslužitelja, te je on započinje isporuku pošte na način da je MX zapis sa najmanjim pripadnim brojem (engl. preference) onaj sa najvećim prioritetom. Klijent tako prolazi listu poslužitelja sve dok ne isporuči e-mail uspješno. Svi poslužitelji koji imaju isti MX broj se tretiraju jednakog prioriteta, pa se stoga nad njima svima iskušava isporuka - dok ne uspije.
- **PTR** (engl. pointer record) - povezuje IPv4 adresu sa odgovarajućim domenskim imenom (FQDN). Obično PTR zapisi trebaju pokazivati na ime koje se može nazad razriješiti u polaznu IPv4 adresu. Naravno, PTR zapis kao takav nije IPv4 adresa, već obrnuto zapisana 4 okteta adrese sa dodatnom IN-ADDR.ARPA. domenom.
- **NS** (engl. name server record) - označava da je za dotičnu zonu treba posluživati upravo dotični DNS poslužitelj. Svaki NS zapis je ili oznaka autoriteta ili oznaka za delegaciju: naime, ako je naziv NS zapisa jednak zoni u kojoj se NS zapis pojavljuje, onda je riječ o autoritativnom zapisu; ako je pak riječ o nazivu koji sadrži neku od poddomena, onda je riječ o delegaciji.
- **SOA** (engl. start of authority) - između ostaloga označava koji je DNS poslužitelj autoritativan za dotičnu domenu, kao i dodatne informacije o zoni. Svaka ispravna zona mora imati SOA zapis.
- **AAAA** i **A6** - povezuju odgovarajuće domensko ime sa IPv6 adresom (128bitna adresa). Moguće je naći i AAAA i A6 zapis, pri čemu se oni razlikuju u nekim detaljima: A6 omogućava da labela bude definirana kao binarni niz, itd. Danas se A6 smatra još uvijek eksperimentalnom, te se preporuča koristiti AAAA u produkciji.
- **DNAME** - relativno recentni način definiranja zamjenskih imena za cijelu domenu, ne nužno samo pojedino domensko ime. Koristi se primjerice u IPv6 za agregaciju i delegaciju cijelog prefiksa. U praksi se rijetko sreće.

- **SRV** (engl. server selection) - zapis koji se sve češće koristi u protokolima koji se tek pojavljuju na tržištu, a predstavlja znatno bolju alternativu trivijalnim MX zapisima. Riječ je o općenitom zapisu za definiciju lokacije servisa, njegove težine i prioriteta, primjerice za LDAP, HTTP, SMTP i sl.
- **TXT** (engl. text string) - pojednostavljeno, omogućava proizvoljan tekstualan zapis do 255 bajtova. Danas se koristi primjerice umjesto zastarjelog **HINFO** opisa uređaja koji nosi domensko ime ili za upisivanje **SPF** (engl. sender policy framework) obilježja.
- **DS** (engl. delegation signer) - dodaje se na mjestu prekida zone (mjestu gdje se vrši delegacija) da bi se pokazalo kako je delegirana zona digitalno potpisana i da dotična prepoznaje određeni ključ kao ispravni vlastiti ključ. Ovime se eksplicitno definira delegacija, umjesto implicitno kao do sada.
- **KEY** (engl. public key) - javni ključ koji je autoriziran od SIG zapisa, a omogućava spremanje i DNSSEC ključeva i proizvoljnih ključeva za aplikacije.
- **KX** (engl. key exchanger) - omogućava metodu za delegiranje autorizacije za neki čvor u ime jednog ili više čvorova, kako bi pružili servise razmjene ključeva.
- **LOC** (engl. location information) - zapis u koji je moguće spremiti geolokacijske odnosno GPS podatke o određenom čvoru ili domeni.
- **SIG** (engl. cryptographic public key signature) - predstavlja potpis radi autentifikacije podataka u DNSSEC-u.
- **TSIG** (engl. transaction signature) - omogućava jednostavnu autentifikaciju koristeći dijeljene tajne ključeve i hashiranje za DNS transakcije.
- **RP** (engl. responsible person) - zapis o odgovornoj osobi za domenu ili čvorove.

Postoji još niz rijetko korištenih zapisa: **AFSDB** (engl. AFS database location code), **HINFO** (engl. host information), **ISDN** (engl. ISDN address), **MB** (engl. mailbox), **MR** (engl. mail rename domain code), **NULL** (engl. null record), **RT** (engl. route through), **X25** (engl. X25 PSDN address), **MINFO** (engl. mailbox or mailing list information), **PX** (engl. pointer to X.400/RFC822 mail mapping information), **NSAP** (engl. network service access point address) i **NAPTR** (engl. naming authority pointer). Praktičnu upotrebu i detaljniji opis pojedinih RR-ova ćemo pokazati kroz primjere u Bind9 poglavlju.

Ponekad možete naići na neke od ovih zapisa, no oni se više ne koriste i preporučljivo ih je izbjegavati koristiti: **WKS** (engl. well known services), **GPOS** (engl. geographical position), **MD** (engl. mail destination), **MF** (engl. mail forwarder), **NSAP-PTR** (engl. NSAP pointer), **NXT** (engl. next domain), itd.

1.9.Primarni i sekundarni NS, prijenos zone

Sa svakodnevnim korištenjem DNS sustava nužno se susresti i sa par dodatnih pojmova i funkcionalnosti, pa počnimo od samog međuodnosa DNS

poslužitelja. Svaki poslužitelj koji ima kompletnu kopiju zone (bilo lokalno bilo prihvatom na neki drugi način) bez potrebe za procesom rezolucije je **autoritativni** DNS poslužitelj za tu zonu. Dakle, riječ je o poslužitelju koji servira vlastite podatke klijentima. Naravno, poslužitelj može biti autoritativan za jednu zonu, ali ne nužno i za neku drugu. Osnovni podatak koji informira poslužitelj da je autoritativan za tu zonu je SOA zapis, uz ostatak konfiguracije koji omogućava prihvrat podataka o zoni i sl. Krivo definirano SOA polje može dovesti do situacije da niti jedan DNS poslužitelj za zonu ne bude autoritativan - i time do prestanka normalnog rada DNS rezolucije za tu zonu.

Može (čak je preporučljivo!) postojati više definiranih DNS poslužitelja za istu zonu koristeći više odgovarajućih NS zapisa. Danas je praksa da bi svaka zona trebala imati barem dva DNS poslužitelja, tako da padom jednog DNS nastavlja funkcionirati - nešto sporije, ali bitno je da su RR-ovi i dalje dostupni. Zašto je bitno imati dva DNS poslužitelja? Nakon isteka TTL vremena pojedinog RR-a (definirano u svakom RR-u), podaci spremljeni po raznim klijentima i poslužiteljima nestaju. U slučaju da je postojao samo jedan autoritativni NS (jedan DNS poslužitelj), a da je on neaktivan ili neispravan - naša zona je nedostupna. I ne samo to - ona je nedostupna na duže vrijeme zbog toga što se neuspjeli upit (NXDOMAIN) spremio na nekom klijentu i njegovom poslužitelju zbog principa negativnog međuspremnika. Stoga je razvijen princip **primarnog** (engl. primary, master) i **sekundarnog** (engl. secondary, slave) DNS poslužitelja. Primarni poslužitelj je onaj autoritativni poslužitelj koji podatke o svojoj zoni ima lokalno spremljeno, odnosno ima im lokalni pristup. Sekundarni poslužitelj je pak onaj koji dobiva podatke od nekog vanjskog izvora, obično koristeći **prijenos zone** (engl. zone transfer) od primarnog poslužitelja. Jasno, primarni poslužitelj za jednu zonu može biti sekundarni za drugu i sl. Sa gledišta klijenta, oba su poslužitelja (primarni i sekundarni) jednake vrijednosti (autoriteta) i jednakog prioriteta (slučajni izbor). Naravno, postoje i drugi razlozi za implementaciju sekundarnog poslužitelja - kako radi lakšeg održavanja (primarni ne mora biti aktivan za vrijeme održavanja), tako i boljeg raspoređivanja opterećenja za velike zone i mnogo upita. Naravno, dobro je i postaviti sekundarni poslužitelj fizički udaljenim iz već navedenih razloga.

Osim primarnih i sekundarnih autoritativnih poslužitelja postoji još par tipova poslužitelja. Počnimo od **isključivo međuspremničkog** poslužitelja (engl. caching-only name server). Takvi poslužitelji nisu autoritativni niti za jedan RR i nemaju nikakve lokalne podatke koje bi posluživali - njihova osnovna funkcija je poboljšati performanse DNS sustava radeći kako pozitivno tako i negativno međuspremanje rezultata DNS upita, smanjujući tako opterećenje na autoritativnim poslužiteljima. Sljedeći tip je **prosljeđivački poslužitelj** (engl. forwarding name server). Njegova je osnovna funkcija prihvrat i prosljeđivanje upita nekom drugom DNS poslužitelju, ali se obično kombinira i sa lokalnim spremanjem dobivenih rezultata - pa je riječ o dobrom rješenju za spore mreže. Još jedan tip je i **isključivo autoritativni** poslužitelj (engl. authoritative-only name server) koji nema međuspremnik DNS upita niti ne odgovara na upite za koje nije autoritativan. On je dakle primarni ili sekundarni poslužitelj za zonu, a ne omogućava rekurzivne upite. Riječ je najčešće o vidu sigurnosti gdje se odvajaju poslužitelji za isključivo

autoritativne i isključivo međuspremničke zadaće. Obično takve okoline gdje se traži sigurna forma DNS poslužitelja imaju nekoliko DNS poslužitelja od kojih su neki javno vidljivi, a neki nisu - pa tvore **skrivenne poslužitelje** (engl. stealth name server). Najčešće je slučaj da skriveni poslužitelji interno isporučuju klijentima DNS informacije koje nisu vidljive na javnoj vanjskoj mreži. Na taj način se vanjskim klijentima poslužuje tek dio informacija za koje se smatra da im je potrebno, a unutrašnjima se daje drugi dio informacija - za koji se smatra da im je dovoljno - i tako se eliminira sigurnosni problem da svi vide "sve". Taj princip se još naziva **razdvojenim poslužiteljima** (engl. split name server), odnosno **razdvojeni DNS** (engl. split DNS).

1.10. Prijenos zone i poboljšanja

Kao što je već rečeno, na primarnom DNS poslužitelju se zona nalazi lokalno, te se i promjene unose lokalno. No, takve podatke nužno je prenijeti na siguran i korektan način do sekundarnih, podređenih DNS poslužitelja i po mogućnosti automatski, odmah nakon završetka uređivanja zone na primarnom. Naime, jednom promijenjeni podaci na primarnom poslužitelju bi bez mehanizma sinhronizacije bili tek djelomično dostupni, budući da se klijentski upiti primarnom i sekundarnim poslužiteljima statistički podjednako raspodjeljuju - pa bi svaki drugi ili n-ti upit za novim zapisom završio ili neuspjehom ili zastarjelim podacima. Nužno je stoga osigurati mehanizme za provjeru svježine podataka na sekundarnom poslužitelju naspram onih na primarnom kao i mehanizme za prijenos zone po potrebi ili barem redovito.

Ključni dio u implementaciji ovih mehanizama je već spomenuti SOA zapis. On sadrži osim podatka tko je autoritativni poslužitelj (i koji je zapravo primarni) za zonu i nekoliko vrlo važnih podataka:

- **Serijski broj** (engl. serial) zone - određuje verziju podataka u zoni, odnosno cijele zone. Pravilo je da se svaki put kad se bilo koji podatak u zoni mijenja, dotični serijski broj mora povećati - bilo automatski (TinyDNS) bilo ručno (Bind). Na taj način se omogućava podređenim poslužiteljima da prepoznaju zastarjelost vlastitih podataka (manji serijski broj - starija zona) i iniciraju prijenos zone. Za serijski broj ne postoje određena pravila, ali se prakticira neki od tri moguća načina: YYYYMMDDn, YYYYMMDDnn i automatski (obično vrijeme promjene zone u sekundama počevši od Epohe). Posljednji broj nn u SOA je u prva dva slučaja redni broj promjene zone unutar dotičnog dana. Nepravilno korištenje SOA polja (primjerice obrnuto korištenje mjeseci MM i dana DD) može uzrokovati desinkronizaciju i zastarjele podatke na sekundarnim poslužiteljima.
- **Vrijeme osvježavanja** (engl. refresh) - označava koliko sekundi će sekundarni poslužitelji čekati između pokušaja osvježavanja zone. Pojednostavljeno, to je najduže vrijeme od promjene zone na primarnom poslužitelju koje je sekundarni čekati prije pokušaja prijena zone.
- **Vrijeme ponovnog pokušaja** (engl. retry) - označava koliko će sekundarni poslužitelji morati čekati nakon neuspješnog prijena zone prije nego pokušaju ponovo. Na ovaj način se jednostavno eliminiraju masovni pokušaji prijena zone koji bi se inače dešavali.

- **Vrijeme isteka** (engl. expire) - definira vrijeme nakon kojeg će sekundarni poslužitelj smatrati vlastite podatke zastarjelima i odbaciti ih, sve do idućeg uspješnog prijenosa zone. Time se jednostavno riješio problem pretjerano zastarjelih zapisa, koji bi unijeli desinkronizaciju u DNS sustav.

Primjer 9: SOA polja u praksi

```
esa.fer.hr          SOA          esa1.esa.fer.hr
postmaster.esa.fer.hr (
                    1124015177      ;serial (version)
                    28800           ;refresh period (8 hours)
                    7200            ;retry interval (2 hours)
                    604800          ;expire time (1 week)
                    604800          ;default ttl (1 week)
                    )
carnet.hr           SOA          dns.carnet.hr
hostmaster.carnet.hr (
                    2005071902      ;serial (version)
                    10800           ;refresh period (3 hours)
                    3600            ;retry interval (1 hour)
                    2419200         ;expire time (4 weeks)
                    86400           ;default ttl (1 day)
                    )
```

Sekundarni poslužitelj po inicijalnom pokretanju može imati bilo nekakvu stariju lokalnu kopiju - koju koristeći SOA polje provjerava naspram primarnog poslužitelja i po potrebi vrši prijenos zone. Naravno, ako nema nikakve podatke, vrši se također prijenos zone.

Sama replikacija podataka, odnosno **prijenos zone** započinje standardnim DNS upitom (dakle UDP) tipa AXFR (engl. address transfer). Na dobiveni zahtjev DNS poslužitelj u slučaju da klijent ima dozvolu odgovara potvrdno, te se klijent ponovno spaja - ovaj put radi pouzdanosti ostvaruje TCP vezu i prenosi čitavu zonu kroz istu vezu, zatvarajući je po završetku. Nakon toga dotični sekundarni poslužitelj odbacuje svoje stare podatke i učitava nove, ponavljajući proces kako je definirano vremenom osvježavanja. Naravno, u slučaju neuspjelog prijenosa također se proces pokušava ponoviti kako je definirano vremenom pokušaja. A ako se desi da prođe vrijeme isteka, odbacuju se svi podaci u sekundarnom poslužitelju sve do prvog uspješnog prijenosa - kao što je već opisano. Naravno, prije nego se obavlja prijenos zone, skoro uvijek se dešava standardni UDP DNS upit za SOA poljem, čime se provjerava da li je zaista prijenos zone potreban - iako je moguće da se taj upit za SOA RR odvija i kroz već uspostavljenu TCP vezu.

Nažalost, koliko god ovaj mehanizam prijenosa bio efikasan sa gledišta jednostavnog polu-automatiziranog prijenosa zone, osnovni problem je da se u praksi u većim organizacijama DNS zone praktički redovno mijenjanju i da je određivanje prijenosa kroz SOA nepraktično - ili je previše rijetko pa se zone ne osvježavaju sukladno sa promjenama, ili je pak previše često - pa se poslužitelj znatno opterećuje velikim i čestim prijenosima. Ono što je

definitivno poboljšanje ovakvog načina je model ugrađen u većinu recentnih DNS poslužitelja:

- Primarni DNS poslužitelj obavještava sve svoje sekundarne poslužitelje o promjeni zone standardnom DNS **porukom obavještenja** odnosno šalje im NOTIFY paket.
- Sekundarni poslužitelji se na prispjeće takve poruke ponašaju kao da im je isteklo vrijeme osvježavanja - te je poboljšanje očigledno: riješio se problem nepotrebnog prozivanja primarnog poslužitelja i skratilo se vrijeme u kojem sekundarni poslužitelji daju zastarjele informacije.

Iduće poboljšanje danas prisutno uglavnom u modernijem DNS softveru poput Bind poslužitelja su **inkrementalni prijenosi zona** (tzv. IXFR) kod kojih se umjesto cijele zone (standardni AXFR) prenosi tek dio promjena, odnosno zadnje promjene. Poslužitelj interno vodi računa o promjenama u lokalnoj zoni: drži lokalnu bazu dotičnih promjena na inkrementalni način, čuvajući razlike između pojedinih verzija. Svaki put kada sekundarni poslužitelj zatraži prijenos zone koristeći IXFR upit (dakle sposoban je za inkrementalni prijenos), poslužitelj iz upita pročitava serijski broj zone koju sekundarni poslužitelj smatra aktualnom i pošalje samo razlike između trenutne i te verzije - odnosno samo promijenjene RR-ove. U praksi se drži tek nekoliko zadnjih verzija zone, pa se u slučaju da primarni poslužitelj nema informacije o nekoj jako staroj zoni vrši puni prijenos. Jasno, u slučaju da primarni poslužitelj ne podržava IXFR ili sekundarni ne šalje IXFR upite, obavlja se isključivo AXFR.

Prijenos zone ima i svoje nedostatke - on nažalost ne garantira nikad da će se prenijeti svi originalni podaci iz zone na primarnom poslužitelju, ali uglavnom se na većini današnjeg DNS softvera prenesu bez problema svi standardni RR-ovi.

1.11. Delegacija

Vratit ćemo se još jednom na netrivialan proces **delegacije**: riječ je o dijeljenju određene zone u podzone, koristeći odgovarajuće NS zapise - u svojem delegacijskom obliku. No, na nekoliko je važnih detalja potrebno obratiti pažnju: ako se zona delegira na DNS poslužitelje čiji je FQDN iz delegirane zone, za normalno funkcioniranje je u matičnoj zoni potrebne definirati odgovarajući **povezujući zapis** (engl. glue records) - A zapis koji definira adrese DNS poslužitelja iz dotične zone. To je nužno zbog toga što se DNS poslužitelji prozivaju po svojim DNS imenima, a ne IP adresama. Da bi se došlo do podataka iz zone, nužno je doći prvo do poslužitelja iz te zone - međutim, u slučaju da ne postoje povezujući zapisi u matičnoj zoni, poslužitelj matične zone ne bi imao dotični podatak te bi jednostavno izdelegirao upit DNS poslužitelju čija se IP adresa još uvijek ne zna.

Nužno je primijetiti kako se svaka promjena autoritativnih poslužitelja za pojedinu domenu (NS zapisi) mora ručno sinkronizirati i na nadređenim poslužiteljima da bi bila očuvana **konzistentnost delegacije** (engl. delegation consistency). U protivnom nema poante postojanja dotičnih poslužitelja koji neće biti dostupni (nemaju povezujuće zapise na matičnoj zoni) jednom kad

prođe TTL za njihove A zapise. Sljedeći čest problem je **kriva delegacija** (engl. lame delegation), kada NS naveden u matičnoj zoni kao autoritativni za zonu ne pruža autoritativne odgovore. Postoji nekoliko razloga za takvo ponašanje: a) nema aktivnog DNS poslužitelja, b) poslužitelj je aktivan ali je bez autoritativnih podataka (svi su istekli, sekundarni, nije bilo recentnog prijenosa zone) ili c) odgovara sa porukom o greški (SERVFAIL ili REFUSED). Dakle, problem je do nekonzistentne definicije delegacije (različiti NS zapisi na matičnoj i delegiranoj zoni) ili do toga da su u obje zone NS-ovi krivo postavljeni (pokazuju na krivi ili loše konfigurirani poslužitelj). Kod postavljanja delegacije nužno je pripaziti da se ne desi **kružna međuovisnost** (engl. cyclic dependancy) kod kojeg jedan dio DNS stabla sadrži međusobne ovisnosti između dvije zone, onemogućujući time normalan rad DNS-a. DNS klijenti standardno mogu prolaziti različitim dijelovima DNS stabla da bi pronašli traženi zapis - no kod međusobne ovisnosti će takvi upiti završiti petljom i nikad se ne dolazi do odgovora.

Završni slučaj delegacije je vjerojatno i najkompliciraniji, međutim pokazuje eleganciju rada sa DNS sustavom. **Delegacija pod mreže bez upotrebe klasa** (engl. classless subnet delegation) je danas odgovor na nužnost delegiranja tek jednog dijela reverzne (IN-ADDR.ARPA) zone. Naime, za upravljanje reverznom zonom standardno se delegirala najmanja mreža, pod mreža klase C sa 256 adresa - što se vrlo brzo pokazalo nepraktičnim zbog velike i nepotrebne potrošnje IP adresa. Osnovni način za formiranje ovakve delegacije je koristiti nekoliko odgovarajućih zapisa u reverznoj matičnoj zoni koja će se delegirati:

- **NS** zapise - za definiranje poslužitelja za pod mrežu,
- **PTR** zapise - koji povezuju definirana kanonička imena prema reverznim adresama,
- **CNAME** zapise - koji omogućavaju definiranje zamjenskih imena kako bi se pojednostavio proces.

Kada je jednom ovako definirano, postoje osnovna dva načina delegacije:

- Nadležno tijelo delegira svaku IP adresu kao D klasu pod mreže sa jednim ili više NS zapisom za svaku IP adresu. Onaj tko prima delegaciju morati imati zonu za svaku IP adresu, SOA, dodatne NS-ove i odgovarajući PTR zapis,
- Alternativno matično tijelo ne mora uopće "stvarno" delegirati, već može koristiti praktički proizvoljan CNAME zapis za svaku reverznu adresu (IP) u svojoj reverznoj zoni, zamjenjujući PTR-ove. Pravilo je da se obično ta labela formira iz IP adrese koja se mijenja, a sufiks mora biti domena kojoj se zapravo "prosljeđuje" upit. Na taj način onaj tko prima delegaciju treba imati samo odgovarajući PTR da bi omogućio da se dotična labela razriješi. Ovo je ujedno i danas najpopularniji sustav delegacije bez klasa.

Primjer 10: Reverzna delegacija bez klasa

```
69.2.53.161.in-addr.arpa CNAME 69.srce.hr.
```

1.12.DNS dodaci i neki detalji

Većina današnjih DNS poslužitelja ima ugrađeni vrlo jednostavni i primitivni mehanizam **kružnog posluživanja** (engl. round robin) za koje se smatra da omogućava jednoliko raspoređivanje opterećenja po određanim adresama. Rečeni funkcionira na sljedeći način: u slučaju da je u odgovoru na zadani upit više RR-ova (jedno pitanje - jedan odgovor, više zapisa), redoslijed RR-ova u odgovoru je pseudoslučajan. Imajući u vidu da tipične aplikacije najčešće koriste samo prvi zapis iz odgovora, dobiva se ponašanje gdje aplikacije svaki put kontaktiraju "drugi" poslužitelj. Algoritmi u samim poslužiteljima uglavnom garantiraju cikličnost, a ponegdje ih je moguće mijenjati ili fino podešavati: npr. novije Bind9 inačice imaju `rrset-order` parametar kojim se može definirati cikličnost ili posve slučajan odabir nad istim RR-ovima.

Dotični mehanizam ima osnovni nedostatak u vidu manjka ikakve provjere da li su zapisi ispravni ili da li je određena adresa uopće dostupna - a kamoli koliko je opterećenje na pojedinoj adresi za koju se pokušava implementirati raspodjeljivanje. Ovo se obično rješava koristeći niske TTL vrijednosti i kakvo sučelje prema DNS poslužitelju koje po potrebi omogućava izbacivanje/ubacivanje zapisa u listu ili njihovu promjenu (radi minimizacije čekanja, opterećenja, detekcije mrtvih poslužitelja, itd). Raspodjela će funkcionirati uglavnom dobro dokle god nema slučajeva da svega jedan upit (ili samo jedno računalo) generira vrlo visoka opterećenja.

Drugi, ne tako očit problem je da kružno posluživanje može uzrokovati da se polazno ime u procesu rezolucije neće nužno dobiti nazad iz odgovarajućeg PTR zapisa. U takvom slučaju će dio SMTP poslužitelja, koji implementira provjeru adrese pretražujući unaprijed i unazad DNS rezolucijom, može odbiti isporučiti poštu. Sve u svemu, ikakva raspodjela opterećenja (a kamoli inteligentna raspodjela) po A zapisima je trenutno suboptimalna - pa se smatra da je budućnost korištenje SRV zapisa koji se tek treba dovoljno proširiti. Alternativa je korištenje podservisa unutar postojećih DNS poslužitelja koji bi na osnovu nekih parametara (stanje udaljenih poslužitelja, npr) predali DNS poslužitelju, a zatim i klijentu zadovoljavajući odgovor.

Primjer 11: Kružno posluživanje

Pokušaj 1:

<code>www.google.com</code>	CNAME	<code>www.l.google.com</code>
<code>www.l.google.com</code>	A	<code>66.249.93.104</code>
<code>www.l.google.com</code>	A	<code>66.249.93.99</code>

Pokušaj 2:

<code>www.google.com</code>	CNAME	<code>www.l.google.com</code>
<code>www.l.google.com</code>	A	<code>66.249.93.99</code>
<code>www.l.google.com</code>	A	<code>66.249.93.104</code>

U DNS zoni pojedinih modernijih DNS poslužitelja moguć je i jedan poseban zapis, takozvani **zamjenski zapis** (engl. wildcard). Riječ je o zapisu koji omogućava da jedan zapis postoji umjesto niza drugih istog tipa, koji bi

pokazivali na isti podatak u istoj zoni. U takvom zapisu se koristi znak "*" u imenu kao jedini znak u labeli. Sam DNS poslužitelj će primijeniti dotični zapis i odgovoriti sa dotičnim sadržajem u slučaju da:

- Nema drugih zapisa koji su precizniji (bolji) odgovor na upit, odnosno onih koji točno odgovaraju upitu,
- Zamjenski zapis se može staviti umjesto grupe labela tako da odgovara na zadani upit (engl. pattern matching).

Pojednostavljeno rečeno, zamjenski zapis će omogućiti da se upiti za inače "nepostojećim" labelama preusmjere na "ispravni" RR.

Naposljetku, spomenimo i **dinamički DNS** (engl. dynamic DNS) na klasični DNS sustav. DNS u početku osmišljen s idejom da se promjene u zonama neće prečesto odvijati - što smo već vidjeli kod problematike razmjene i sinkronizacije zona. Za unos u DNS sustav su uglavnom predviđene statičke adrese koje se ne mijenjaju, budući da bi ručno mijenjanje svaki put predstavljalo noćnu moru za održavanje. Moderni DNS i DHCP poslužitelji stoga omogućavaju međusobno povezivanje sustava dodjeljivanja IP adresa sa DNS sustavom, tako da se svako DHCP-registrirano računalo registrira u DNS sustavu kroz automatizirani proces. Specifično, DHCP klijent šalje DNS UPDATE poruku koja indicira DNS poslužitelju što treba obaviti sa odgovarajućim RR-ovima. Naravno, dinamički DNS kao takav nije ograničen nužno na DHCP, već u praksi svaki autorizirani DDNS (dinamički DNS) klijent može upravljati odgovarajućim zapisima u zoni.

1.13.DNS sigurnost

Nažalost, uz DNS sustav su vezani i različiti sigurnosni problemi. Postoji niz trikova pomoću kojih se može odrediti DNS poslužitelj natjerati da prihvati lažne zapise. Takvom metodom **lažiranja DNS zapisa** (engl. DNS forgery) nesvjesni se klijenti preusmjeruju na lažne adrese i time postaju laka meta napadača. Standardno su takvi napadi u formi **trovanja DNS međuspremnik** (engl. cache poisoning), napada kod kojeg se utiče na DNS poslužitelj da povjeruje da je dobio autoritativne informacije o nekim RR-ovima. Time se utiče na sve klijente koji koriste dotični DNS poslužitelj da također koriste lažiranu informaciju, koja može omogućiti daljnje različite napade na klijentska računala.

Postoje tri osnovna tipa ovakvog napada:

- Preusmjeravanje poslužitelja za odredišnu domenu - gdje se za neku domenu na zloćudnom poslužitelju specificira vlastiti NS za traženu domenu u autoritativnom odjeljku i još u dodatnom odjeljku daje vlastiti A zapis sa lažnim NS-om koji se nazivno nalazi u napadnutoj domeni. Zatrovani poslužitelj pamti IP adresu NS poslužitelja koji je sada napadačev DNS poslužitelj i time napadač dobiva mogućnost proizvoljnog baratanja sa cijelom napadnutom zonom.
- Preusmjeravanje NS zapisa odredišne domene - omogućava preusmjeravanje DNS poslužitelja neke druge domene (nevezane uz originalni upit) na proizvoljnu napadačevu IP adresu. Napadačev DNS poslužitelj odgovara u autoritativnom odjeljku za napadnutu domenu (nevezanu uz originalni upit) sa NS zapisom u traženoj domeni, a u

dodatnom odgovoru daje A zapis sa IP adresom dotičnog DNS poslužitelja. Time dolazi do iste funkcionalnosti kao i u prošlom napadu.

- Treći tip napada je napad identifikacijom - kod kojeg je osnovna ideja predviđanje 16bitnog identifikacijskog broja u DNS komunikaciji. Ako napadač uspješno pogodi isti i bude prvi koji vraća odgovor sa ispravnim brojem, poslužitelj/klijent će tretirati njegov odgovor kao ispravan i autoritativan. Nažalost, sa što većim brojem istovremenih DNS upita koje poslužitelj obrađuje, vjerojatnost uspješnog pogađanja (odnosno vjerojatnost kolizije) jedinstvenog broja upita se povećava. Danas moderni softver uglavnom taj problem rješava kvalitetnijim pseudo-slučajnim generatorima kao i slučajnim izborom visokih izvorišnih portova za upite (budući da odgovor mora biti poslan na isti izvorišni port).

Većina ovih napada danas je riješena promjenama u DNS softveru (dakle noviji Bind9 i Djbdns softver) koji uglavnom ignorira dobivene DNS odgovore koji nisu striktno vezani uz prvotni zadani upit. Jedno od osnovnih mjera zaštite je ograničenje rekurzivnih upita isključivo na područje lokalne mreže. U praksi je ovo česta pogreška, budući da Bind poslužitelj o osnovnoj konfiguraciji omogućava rekurzivne upite svima - pa je time udaljenom napadaču cijela procedura trovanja međuspremnik nažalost jednostavnija za izvedbu.

Alternativni i sve popularniji pristup sigurnosti je uvođenje sigurnog DNS-a, tzv. **DNSSEC** sustava. Pojednostavljeno, riječ je o korištenju odgovarajućih RR-ova za potpisivanje dijelova zona ili čak cijele komunikacije koristeći digitalne potpise i digitalne certifikate kako bi se potvrdila izvornost, integritet i autentičnost DNS podataka. Na taj način (provjeravajući potpis i podatke u zoni) DNS klijent može provjeriti podatke i za sigurnošću znati jesu li oni zaista potekli od traženog autoritativnog DNS poslužitelja.

Naposljetku spomenimo problem ne toliko vezan uz sigurnost koliko vezan uz **DNS zagađenje** (engl. DNS pollution) odnosno bespotrebne DNS upite. primjerice DNS upite za RFC 1918 privatnim adresama je obično dobro lokalno terminirati na DNS poslužitelju. Takav promet bespotrebno opterećuje vršne DNS poslužitelje kao i vaš poslužitelj - budući da se takve adrese koriste isključivo u privatnim mrežama te niti jedan DNS poslužitelj u svijetu neće biti autoritativan za rečene adrese. Terminacija se obično rješava tako da se na razini DNS poslužitelja definiraju "prazne" reverzne zone za RFC 1918 klase: 10.in-addr.arpa, 16.172.in-addr.arpa do 31.172.in-addr.arpa i 168.192.in-addr.arpa. Prema recentnim istraživanjima oko 7% svjetskog DNS prometa predstavlja curenje RFC 1918 upita prema vršnim DNS poslužiteljima, stoga je 2002. godine formirana dodatna usmjerivačka hijerarhija oko AS112 radi terminiranja upita za RFC 1918 (10.in-addr.arpa, itd.) i RFC 3330 (254.169.in-addr.arpa) adresama.

Postoje još različiti tipovi zagađenja koja se dešavaju u DNS prostoru:

- A-A upiti - neispravni DNS klijent šalje A upit u kojem je već sadržana IP adresa ("Koja je IP adresa računala sa IP adresom 1.2.3.4?"). Ovo

je karakteristično za Microsoft Windows NT operacijski sustav, a rješava se obično korištenjem djbdns servisa ili Bind9 poslužitelja koji je autoritativan za svih 256 numeričkih zona, pri čemu je svaka prazna.

- Upiti za krivim TLD-ovima - koji su najčešće greška u lokalnim konfiguracijama (kriva domena, netočna domena, mobilni uređaji, neispravne standardne konfiguracije) ili aplikacijama, pa cure upiti za "localhost", "localdomain", "workgroup" i sličnim nepostojećim domenama, odnosno domenama koje bi trebale biti lokalno definirane.
- Upiti za adresama vršnih poslužitelja - tipično svi DNS poslužitelji imaju popis vršnih poslužitelja kako bi uopće mogli ostvariti početnu komunikaciju. Povremeno osvježanje zapisa je normalno zbog istjecanja TTL-a, no RR-ovi za vršne poslužitelje imaju tipično TTL od 1000 sati. Ovo je također najčešće greška u filtriranju DNS prometa, neispravnom DNS poslužiteljskom softveru i sl.
- IPv6 upiti - Bind poslužitelj tipično dodatno obavlja najčešće nepotrebne (čak i ako računalo nema IPv6 stog) AAAA i A6 upite, primjerice za povezujuće zapise.

2. DNS alati

Postoji cijeli spektar različitih alata kako za iskusnog DNS administratora, tako i za početnika. Stoga donosimo tek osnovne alate koji bi trebali omogućiti testiranje individualnih zapisa, konfiguracija i cijelih zona. Nadalje, nekad mnogo korištenu naredbu `nslookup` ne spominjemo iz jednostavnog razloga - neoprostivo je zastarjela i praktički neupotrebljiva za iole složenije zadaće.

2.1.Naredba *host*

Nažalost postoje dvije inačice ove naredbe sa istim imenima - jedna je ona koju donosi Bind9 programski paket, a druga je slobodno dobavljiva i nalazi standardno se u većini različitih Unixoida i Linux distribucija. Mi ćemo se ovdje orijentirati na potonju inačicu, koja ima prilično više mogućnosti i dodataka. Osnovna sintaksa naredbe je sljedeća:

```
host [-v] [-a] [-t tip_upita] [parametri] [poslužitelj]
host [-v] [-a] [-t tip_upita] [parametri] -l zona
[poslužitelj]
```

Argumenti naredbi su sljedeći:

- `-v` daje kompletne informacije pri pregledu RR-ova (TTL, klase), te sve odjeljke (dodatni i autoritativni),
- `-t` parametar omogućava pretragu za proizvoljnim tipom RR (moguće je zadati sve tipove koje smo već spomenuli),
- `-a` odgovara `-t any` (odnosno `-t *`),
- `-l` omogućava pregled svih zapisa u zoni (obavlja AXFR), te je sa `-t` moguće filtrirati koje specifične tipove RR-ova se traži iz cijele zone,
- `-p` pri ispisu zone forsira da se obavlja prijenos zone samo sa primarnog poslužitelja,
- `-d` omogućava još detaljniji ispis sa prikazom komunikacije i grešaka,
- `-Z` daje ispis kakav odgovara standardnoj Bind zoni.

Primjer 12: Korištenje naredbe *host*

Ispis DNS poslužitelja za `carnet.hr` domenu u Bind formatu:

```
$ host -Z -t ns carnet.hr
carnet.hr.                20667      IN         NS
dns.carnet.hr.            20667      IN         NS
carnet.hr.                20667      IN         NS
dns2.carnet.hr.           20667      IN         NS
carnet.hr.                20667      IN         NS
bjesomar.srce.hr.
```

Ispis TXT polja za `fsb.hr` domenu:

```
$ host -t txt fsb.hr
fsb.hr                    TXT        "v=spf1
ip4:161.53.116.0/22 ip4:193.198.206.0/24
ip4:193.198.217.192/27 a mx ptr ~all"
```

Ispis svih A zapisa u bofhlet.net domeni:

```
$ host -l -t a bofhlet.net
bofhlet.net.      A      38.119.119.63
ftp.bofhlet.net.  A      38.119.119.63
host.bofhlet.net. A      38.119.119.63
localhost.bofhlet.net. A      127.0.0.1
```

Pregled DNS poslužitelja za hr ccTLD preko dns.srce.hr poslužitelja (primijetite točku na kraju domene):

```
$ host -v -t ns hr. dns.srce.hr
Server: dns.srce.hr
Address: 161.53.3.7
```

Query about hr. for record types NS

Trying hr ...

Query done, 5 answers, authoritative status: no error

hr	86400	IN	NS	
sunic.sunet.se				
hr	86400	IN	NS	ns-
ext.vix.com				
hr	86400	IN	NS	ns.uu.net
hr	86400	IN	NS	
dns.srce.hr				
hr	86400	IN	NS	
ns1.univie.ac.at				
Additional information:				
ns.uu.net	3594	IN	A	
137.39.1.3				
dns.srce.hr	86400	IN	A	
161.53.3.7				
ns1.univie.ac.at	68394	IN	A	
193.171.255.2				
sunic.sunet.se	86394	IN	A	
192.36.125.2				
ns-ext.vix.com	3594	IN	A	
204.152.184.64				
ns-ext.vix.com	3594	IN	AAAA	
2001:4F8:0:2:0:0:0:13				

2.2.Naredba dig

Naredba `dig` je pripadnik nešto starije generacije programa, pa je dobar dio njegove funkcionalnosti pokriven u `host` naredbi. No njegova jednostavna sintaksa je prednost za većinu DNS administratora, a i standardno generira potpuni ispis nalik na Bind zonu. Također, rečeni alat ima niz korisnih zastavica za detekciju i otklanjanje grešaka u udaljenoj konfiguraciji. Najjednostavniji način upotrebe naredbe `dig` je sljedeći:

```
dig @poslužitelj ime_zapisa tip_zapisa
```

Pri čemu je poslužitelj u formi IPv4 ili IPv6 adrese, ime zapisa je traženo ime RR-a, a tip je odgovarajući tip RR-a. Standardno `dig` ispisuje sve komentare, koje je moguće ugasiti korištenjem parametra `+nocomments`. Također spomenimo par korisnijih odnosno češće korištenih parametara:

- `+tcp`, `+notcp` - forsiraju korištenje TCP odnosno UDP DNS komunikacije,
- `+search`, `+nosearch` - omogućuju odnosno onemogućuju čitanje `/etc/resolv.conf` za `search` te domain parametrima,
- `+recurse`, `+norecurse` - omogućuje odnosno onemogućuje postavljanje RD zastavice odnosno rekurzije udaljenog poslužitelja,
- `+trace`, `+notrace` - pregled svih izvršenih upita da se zadovolji pretraga počevši od korijenskih DNS poslužitelja nadalje.

Primjer 13: Korištenje naredbe `dig`

Ispišimo A zapis za `jagor.srce.hr`:

```
$ dig jagor.srce.hr
```

```
; <<>> DiG 9.2.4 <<>> jagor.srce.hr
;; global options: printcmd
;; Got answer:
;; ->>HEADER<- opcode: QUERY, status: NOERROR, id: 60167
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2,
ADDITIONAL: 2

;; QUESTION SECTION:
jagor.srce.hr.                IN      A

;; ANSWER SECTION:
jagor.srce.hr.                86400   IN      A
161.53.2.130

;; AUTHORITY SECTION:
srce.hr.                      86400   IN      NS
bjesomar.srce.hr.
srce.hr.                      86400   IN      NS
regoc.srce.hr.

;; ADDITIONAL SECTION:
regoc.srce.hr.                86400   IN      A
161.53.2.69
bjesomar.srce.hr.            86400   IN      A
161.53.2.70

;; Query time: 0 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Sun Aug 21 16:37:49 2005
;; MSG SIZE rcvd: 122
```

Ispišimo svih 13 korijenskih DNS poslužitelja:

```
$ dig +nocomments ns . @a.root-servers.net.
; <<>> DiG 9.2.4 <<>> +nocomments ns . @a.root-
servers.net.
;; global options: printcmd
;.                               IN      NS
.                               518400 IN      NS      L.ROOT-
SERVERS.NET.
.                               518400 IN      NS      M.ROOT-
SERVERS.NET.
itd.
```

2.3.Naredba *dnswalk*

Jednom kad su postavljene zone i kad ih DNS poslužitelj servira svojim klijentima, poželjno je redovno provjeravati ispravnost istih. Jedan od jednostavnijih alata za ovu namjenu je *dnswalk*, koji će koristeći AXFR preuzeti željenu zonu i ispisati različite utvrđene nekonzistentnosti iste. Prije upotrebe nužno je osigurati se da sa računala klijenta imate dozvole za prijenos zone (AXFR). Sintaksa je jednostavna (primijetite obaveznu točku na kraju domene):

```
dnswalk [ -adilrffm ] domena.
```

Parametri imaju sljedeća značenja:

- *-a* upozorava na višestruke A zapise,
- *-r* rekurzivno silazi po poddomenama u potrazi za greškama,
- *-d* ispisuje greške na standardni izlaz za greške,
- *-m* provjerava zonu samo ako je promijenjena od zadnjeg pokretanja ovog programa,
- *-F* provjerava adrese tako da radi standardnu rezoluciju pojedinog A zapisa i provjerava dobiveni izlaz sa rekurzivnom rezolucijom (PTR) i uspoređuje dobivene rezultate,
- *-i* onemogućuje provjeru za krivim znakovima u labelama,
- *-l* omogućava provjeru za neispravnim delegiranjem.

Primjer 14: Korištenje naredbe *dnswalk*

Pregled grešaka u *fsb.hr* domeni:

```
$ dnswalk -Falr fsb.hr.
Checking fsb.hr.
Getting zone transfer of fsb.hr. from
hobbit.fsb.hr...done.
SOA=localhost.fsb.hr      contact=postmaster.fsb.hr
WARN: www.coma.fsb.hr CNAME zrno.fsb.hr: CNAME (to
karmela.fsb.hr)
WARN: www.zrno.fsb.hr CNAME zrno.fsb.hr: CNAME (to
karmela.fsb.hr)
0 failures, 2 warnings, 0 errors.
```

2.4.Naredba *fpdns*

Naredba *fpdns* kao osnovnu namjenu detekcija softvera udaljenog DNS poslužitelja. Ovo je u praksi vrlo korisno za eliminiranje potencijalnih problema u komunikaciji (primjerice, između Djbdns i Bind9 poslužitelja i sl). Naravno, kao osnovna metoda detekcije na većini Bind poslužitelja može poslužiti i naredba *host*:

```
host -t txt -c chaos version.bind poslužitelj
```

No, za skoro sve druge poslužitelje nema neke općeprihvaćene i standardne metode - pa stoga *fpdns* vrši različite specifične upite i uspoređuje prema internoj bazi za svaki softver. Također, *fpdns* ukazuje na omogućenost rekurzivnih upita udaljenom klijentu, što je također vrlo važan dijagnostički podatak: u slučaju da udaljeni DNS poslužitelj omogućava rekurziju klijentu koji nije u njegovoj domeni očigledno je riječ o ozbiljnoj ranjivosti. Događajna naredba ima sljedeću sintaksu:

```
fpdns.pl [ -c ] [ -d ] [ -f ] [ -F broj_djece ] [ -p port ]  
[ -Q izvorišna_adresa ] [ -r broj_pokušaja ] [ -s ]  
[ -t vrijeme_upita ] [ -v ] [ poslužitelj ]
```

Parametri imaju ova značenja:

- *-c* koristi pregled CH TXT polja (za Bind softver) ako je moguće, no to se standardno ne podrazumijeva zbog nepreciznosti (administrator može upisati proizvoljan sadržaj);
- *-d* omogućava detaljni ispis grešaka i komunikacije,
- *-f* forsira upotrebu CH TXT,
- *-F* omogućava kontrolu djece-procesa koji obavljaju identifikaciju, standardno je to 10 primjeraka,
- *-p* daje mogućnost promjene odredišnog porta za DNS komunikaciju - naravno, to je standardno port 53,
- *-Q* omogućava izbor izvorišne adrese što je korisno primjerice na računalima sa više mrežnih adresa,
- *-r* kontrolira broj ponovnih pokušaja identifikacije,
- *-s* smanjuje izlazni ispis na što manje informacija,
- *-t* omogućava kontrolu ukupnog vremena komunikacije - primjerice da se ne čeka na poslužitelj koji je nedostupan.

Primjer 15: Korištenje naredbe *fpdns*

Saznavanje verzije poslužitelja dns.carnet.hr:

```
$ fpdns dns.carnet.hr  
fingerprint (dns.carnet.hr, 161.53.123.3): BIND 9.2.0rc7  
-- 9.2.2-P3 [recursion enabled]
```

Odnosno verzije esa1.esa.fer.hr poslužitelja:

```
$ fpdns esa1.esa.fer.hr
```

```
fingerprint (esa1.esa.fer.hr, 161.53.71.194): TinyDNS
1.05
```

Te verzije DNS poslužitelja za google.com domenu:

```
$ fpdns ns1.google.com
fingerprint (ns1.google.com, 216.239.32.10): BIND 8.3.0-
RC1 -- 8.4.4
```

2.5.Naredba *nslint*

Za razliku od većine opisanih alata, *nslint* je naredba koja lokalno provjerava ispravnost Bind zona. Na ovaj način možete provjeriti većinu problema prije nego se uopće počnu servirati potencijalno neispravne zone. Nažalost, ovaj program ima minus što prijavljuje pretjerano mnogo različitih informacija o potencijalnim (dakle ne i stvarnim) problemima koje nije moguće filtrirati ili kontrolirati. Također, budući da ovaj program radi isključivo lokalno - ograničen je na Bind zone. Sintaksa je trivijalna, sa *-c* parametrom se prosljeđuje put do *named.conf* konfiguracijske datoteke za Bind poslužitelj.

Primjer 16: Korištenje naredbe *nslint*

```
$ nslint -c /etc/named.conf
nslint: 161.53.116.6 in use by frodo.fsb.hr. and
*.hsasf.hr.
nslint: 161.53.116.6 in use by hsasf.hr. and
frodo.fsb.hr.
nslint: 161.53.116.6 in use by www.hsasf.hr. and
hsasf.hr.
itd.
```

2.6.Naredba *zonecheck*

Ovaj alat je za krajnjeg korisnika još jednostavniji od *dnswalk* naredbe, a obavlja daleko više temeljitih pretraga ispravnosti i konzistencije DNS zona, kao i različitih preduvjeta za ispravno funkcioniranje. Izvještaji su jasni i pregledni, uz svaki problem je priloženo i vrlo jasno objašnjenje na engleskom jeziku - pa je time i dobro za DNS administratora-početnika. Uz već spomenute prednosti, *zonecheck* ima i niz argumenata koji omogućavaju fino upravljanje nad ispisom i testovima. Sintaksa je sljedeća:

```
zonecheck [ -hqV ] [ -voet opt ] [ -46 ] [ -c
konfiguracija ] [ -n lista_poslužitelja ] domena
```

Argumenti su mnogobrojni ali ne i nužni za normalan rad, gdje su standardne postavke dovoljno dobre. Stoga nećemo ići u detalje, a zainteresiranima preporučujemo čitanje odgovarajućih priručnika uz program.

Primjer 17: Korištenje naredbe *zonecheck*

Provjera bofhlet.net domene:


```
$ zonecheck bofhlet.net
ZONE : bofhlet.net.
NS <= : ns1.bofhlet.net. [38.119.119.63]
NS : ns2.bofhlet.net. [38.119.119.64]

      ,------.|
~~~~~|   warning   ||
~~~~~|-----'
      \-----'

w> Nameservers are all part of the same AS
  | Adv: ZoneCheck
  |   To avoid loosing all connectivity with the
authoritative DNS in case
  | of a routing problem inside your Autonomous System, it
is advised to
  | host the DNS on different AS.
  \----- - - - - -
      : All the nameservers are part of the same Autonomous
System (AS number
      : 21792), try to have some of them hosted on another AS.
      \..... .. . . .
=> generic

w> Reverse for the nameserver IP address doesn't match
=> ns2.bofhlet.net./38.119.119.64
=> ns1.bofhlet.net./38.119.119.63

==> SUCCESS (but 3 warning(s))
```

2.7.Naredba dnstop

Riječ je o servisu koji omogućava jasne i pregledne statistike DNS upita u stvarnom vremenu. Rečeni servis prisluškuje mrežni promet i analizira DNS upite, kategorizirajući ih po: domeni (TLD, drugog i trećeg stupnja), tipu upita, tipu operacije te izvorišnoj ili odredišnoj adresi. Također postoji par ugrađenih filtera koji omogućavaju pregled tipičnih zagađenja DNS-a: A-A upite i upite za nepostojećim domenama. Sintaksa je sljedeća:

```
dnstop [-aps] [-b izraz] [-i adresa] [-f filter] [uređaj]
[datoteka]
```

Parametri imaju ova značenja:

- -a anonimizira odnosno skriva adrese u ispisu,
- -b definira BPF izraz za filtriranje mrežnog prometa (tipično je sljedeći: `udp dst port 53 and udp[10:2] & 0x8000 = 0`),
- -i omogućava ignoriranje rečene adrese,
- -o onemogućava postavljanje mrežnog sučelja u način za prisluškivanje svog prometa (engl. promiscuous mode),

- `-s` i `-p` omogućavaju prikupljanje informacija o domenama drugog i trećeg stupnja,
- `-f` omogućava korištenje dodatnih filtara DNS upita; na raspolaganju su `unknown-tlds` za pregled upita za nepoznatim vršnim domenama, `A-for-A` filtar za pregled računala koja šalju nepotrebne A-A upite i `rfc1918-ptr` filtar za pregledom računala koja šalju upite za RFC 1918 privatnim adresama,
- `uredaj` definira mrežno sučelje na kojem će se obavljati prisluškivanje (`eth0`, `fxp0`, `ee0`),
- `datoteka` definira datoteku u kojoj se nalazi već spremljen mrežni promet za analizu.

Kako je program interaktivan, na raspolaganju su različite tipke koje mijenjaju prikaz statistika uživo:

- `s` prikazuje tablicu sa statistikama izvorišnih adresa,
- `d` prikazuje tablicu sa statistikama odredišnih adresa,
- `t` prikazuje statistiku tipova upita,
- `o` prikazuje statistiku tipova operacija,
- `1` prikazuje TLD tablicu,
- `2` prikazuje SLD (domene drugog stupnja) tablicu,
- `3` prikazuje 3LD (domene trećeg stupnja) tablicu,
- `c` prikazuje SLD i izvorišnu tablicu table,
- `#` prikazuje 3LD i izvorišnu tablicu,
- `^R` resetira brojače,
- `^X` izlazi iz programa,
- `^C` izlazi iz programa,
- `?` je interaktivna pomoć.

3. Bind9 poslužitelj

Danas najrašireniji DNS poslužiteljski softver je ISC Bind. Danas svi vršni DNS poslužitelji koriste upravo Bind softver, pretežno u verziji 8 - pa je riječ o praktičnom standardu. Nažalost, ovaj softver prati loš glas, budući da je u svojoj prošlosti (Bind 4 i Bind 8 verzije) imao niz kritičnih sigurnosnih propusta. Nažalost, i dan danas se pojavljuju kritične rupe u ovim inačicama softvera. Verzija 9 je navodno napisana u potpunosti iz početka, te je riječ o softveru koji podržava do danas najveći broj DNS specifikacija i dodataka - DNSSEC, TSIG, DDNS, NOTIFY mehanizam, EDNS0, IXFR, IPv6, itd. Riječ je o vrlo naprednom servisu koji je specifično pisan sa višeprocorskom i višedretvenom podrškom na umu, udaljenom kontrolom kroz `rndc` program, mogućnošću chrootanja i sl. Riječ je o moćnom DNS poslužitelju koji omogućava implementaciju najsloženijih zadataka - ali koji nije nužno najsigurniji ili baš nužan za jednostavnije zadatke.

3.1. Konfiguracija općenito

Osnovna konfiguracija servisa je datoteka `named.conf`. Ona se obično nalazi u `/etc` ili specifično za Debian u `/etc/bind` direktoriju. Popularna je i varijanta sa spremanjem u `/etc/namedb` direktorij. Sama konfiguracijska datoteka za servis se sastoji od nekoliko grupa direktiva odnosno odjeljaka (donosimo najvažnije, a izostavljamo `masters`, `trusted-keys` i `lwres` odjeljke)

- **Komentari** - koji omogućavaju da dio konfiguracijske datoteke (linija po linija ili čak grupa linija) bude zanemaren pri učitavanju servisa. Upotreba komentara u konfiguraciji je esencijalna za komplicirane konfiguracije budući da omogućava administratorima jasan pregled kad i zašto je što promijenjeno, koji dio konfiguracije čemu služi i za slične namjene,
- **Pristupne liste** (`acl` direktiva) - pristupne liste adresa ili korisnika za određenu primjenu kasnije u ostalim konfiguracijskim direktivama. Korištenje pristupnih lista je isključivo radi olakšanja kasnijih definicija dozvola u ostalim blokovima, pa se njihova upotreba preporuča radi bolje preglednosti,
- **Ključevi** (`key` direktiva) - koji omogućavaju autorizaciju za određenu zonu (primjerice za DDNS i DNSSEC) ili ključeve za udaljeno upravljanje kroz `rndc` program. U standardnoj upotrebi ovaj odjeljak nije potreban,
- **Server grupa** (`server` direktiva) - kroz koju se definira ponašanje DNS poslužitelja u odnosu na druge poslužitelje i klijente (prijenosi zona, koji portovi se koriste, itd). U standardnoj upotrebi ovaj blok nije potreban, ali omogućava da se promijeni ponašanje samo za kakav specifičan poslužitelj,
- **Kontrole** (`controls` direktiva) - služe definiranju dozvola udaljenog upravljanja servisom kroz `rndc` program. Standardno ni ovaj odjeljak nije nužan, te ga je moguće izostaviti,

- **Zapisnici** (`logging` direktiva) - služi definiranju mjesta, nivoa i tipa spremanja poruka o radu servisa. Standardno su definirani svi potrebni servisi i koriste se sistemski zapisnici, pa je za svakodnevnu upotrebu moguće izostaviti cijeli odjeljak,
- **Parametri rada** (`options` direktiva) - niz parametara koji određuje ponašanje cijelog servisa i svih zona. Ovaj odjeljak je prilično bitan i postoji cijeli niz opcija koje je preporučljivo podesiti - posebice ako DNS poslužitelj ima nekoliko mrežnih adresa,
- **Pogled** (`view` direktiva) - omogućava podešavanje različitih pogleda i ponašanja zona i te promjena serviranja DNS informacija u ovisnosti o zadanim kriterijima. Ovaj odjeljak je bitan isključivo ako se planira razdvojiti DNS poslužitelj na unutrašnji i vanjski odnosno omogućiti da različiti klijenti vide istu zonu na različite načine,
- **Umetnuta datoteka** (`include` direktiva) - omogućava da se umeće dio konfiguracije iz neke druge datoteke, a dobiveni sadržaj se tretira kao da je jedna jedinstvena datoteka. Za manje poslužitelje korištenje dijelova konfiguracije iz drugih datoteka obično samo komplicira održavanje, budući da je konfiguracija onda raspršena na nekoliko datoteka,
- **Zone** (`zone` direktiva) - definira zone koje će poslužitelj posluživati klijentima. Iznimno bitan odjeljak, kojeg svaki standardni tip poslužitelja mora imati.

3.2.Komentari

Dozvoljeni komentari u `named.conf` datoteci su vrlo slobodno definirani, za razliku od onih dozvoljenih u zonama. Specifično, moguće je koristiti (naravno, bez navodnika):

- C komentare: počinju sa `/*` i završavaju sa `*/`, a mogu se protezati kroz nekoliko redova,
- C++ komentare: počinju sa `/**` i vrijede do kraja tekuće linije,
- Unix komentare: počinju sa `#` i vrijede do kraja tekuće linije.

Primjer 18: Komentari u `named.conf` datoteci

```
/* ovo je komentar
...niz komentara, potencijalno kroz više redova...
ali komentar završava sa ovim */
# komentar kroz jedan red
// ili ovakav komentar kroz jedan red
```

3.3.Parametri rada servisa

Dotična `options` direktiva mijenja globalno ponašanje cijelog servisa, a ima slijedeću sintaksu. Primijetite da spominjemo samo najvažnije i najčešće parametre, pa ova lista nije ni izdaleka potpuna, ali će zadovoljiti većinu standardnih potreba. Također, iz verzije u verziju Bind9 poslužitelja se pojavljuju i neki dodatni napredni parametri (`rrset-order` primjerice):

```
options {  
    [ version version_string; ]  
    [ directory path_name; ]  
    [ minimal-responses yes_or_no; ]  
    [ notify yes_or_no | explicit; ]  
    [ recursion yes_or_no; ]  
    [ forward ( only | first ); ]  
    [ forwarders { ip_addr [port ip_port] ; [ ip_addr  
[port ip_port] ; ... ] }; ]  
    [ allow-notify { address_match_list }; ]  
    [ allow-query { address_match_list }; ]  
    [ allow-transfer { address_match_list }; ]  
    [ allow-recursion { address_match_list }; ]  
    [ blackhole { address_match_list }; ]  
    [ listen-on [ port ip_port ]  
{ address_match_list }; ]  
    [ query-source [ address ( ip_addr | * ) ] [ port  
( ip_port | * ) ]; ]  
    [ transfer-format ( one-answer | many-answers ); ]  
    [ transfer-source (ip4_addr | *) [port ip_port] ; ]  
    [ notify-source (ip4_addr | *) [port ip_port] ; ]  
    [ provide-ixfr yes_or_no; ]  
    [ request-ixfr yes_or_no; ]  
    [ port ip_port; ]  
};
```

Krenimo redom:

- `version` - omogućava promjenu već spomenute CH TXT verzije Bind poslužitelja, pa se najčešće koristi sa skrivanje stvarne verzije poslužitelja radi nekakve lažne sigurnosti,
- `directory` - postavlja radni direktorij u odredišni, tako da je za zone i ostale direktive gdje se otvaraju datoteke moguće koristiti relativne staze,
- `minimal-responses` - servis će dodavati autoritativni i dodatni odjeljak u DNS odgovore samo kad je to nužno (za delegacije i negativne odgovore), što obično poboljšava performanse DNS poslužitelja (standardno ova opcija nije postavljena),
- `notify` - poslužitelj šalje NOTIFY poruke svim poslužiteljima u NS zapisima za zonu (osim onom iz SOA polja) kad se desi promjena zone na autoritativnom DNS poslužitelju (standardno postavljena),
- `recursion` - poslužitelj će obaviti sav standardni posao oko odgovora na rekurzivne upite; u protivnom, poslužitelj će odgovarati samo iterativnim odgovorima, dakle ili proslijeđivši dalje ili iz lokalnog DNS međuspremnik; ovu je opciju poželjno staviti samo za računala iz vlastite lokalne mreže (standardno postavljeno),
- `forwarders` - definira listu poslužitelja za proslijeđivanje DNS upita (standardno se upiti ne proslijeđuju),

- `forward` - omogućuje da se upiti isključivo proslijeđuju (`forward only`) ili da se prvo proslijedi pa obavlja normalan tip upita (`forward first`) (standardno se upiti ne proslijeđuju),
- `allow-notify` - definira kojim je poslužiteljima dozvoljeno da šalju NOTIFY poruke (standardno se primaju poruke samo od primarnog poslužitelja za zonu),
- `allow-transfer` - definira kojim je poslužiteljima omogućen prijenos svih zona; za ovu je opciju poželjno postaviti isključivo nadređene i podređene DNS poslužitelje (standardno je dozvoljen prijenos svim računalima - ključna riječ `any`),
- `allow-recursion` - definira za koja računala poslužitelj smije obavljati rekurzivne upite; poželjno je postaviti samo računala iz lokalne mreže (standardno je dozvoljeno svima),
- `blackhole` - definira listu adresa sa kojih poslužitelj neće prihvaćati nikakve upite, niti će im odgovarati (standardno je ova lista prazna - ključna riječ `none`),
- `listen-on` - definira portove i adrese na kojima će poslužitelj osluškivati za upitima; poželjno je pripaziti na ovu konfiguraciju kod računala sa više mrežnih adresa, budući da je poželjno da DNS poslužitelj obično osluškuje na poznatoj javnoj adresi (standardno port 53, sve raspoložive adrese na lokalnom računalu),
- `query-source` - definira port i adresu sa kojeg će poslužitelj slati daljnje upite; također je poželjno ovo postaviti na točno određenu adresu kod računala sa više mrežnih adresa (standardno se koristi bilo koja adresa lokalnog računala i bilo koji visoki port),
- `transfer-format` - omogućava promjenu oblika prijenosa zone, pa `one-answer` koristi jednu DNS poruku po jednom RR, dok `many-answer` pakira što više RR-ova u jednu poruku; u slučaju problema u prijenosu zone sa starim DNS poslužiteljima potrebno je promijeniti ovaj parametar (standardno se koristi `many-answer`),
- `transfer-source` - definira koja se lokalna adresa koristi za izvorišnu adresu kod prijenosa zone; ovo je posebice korisno kod računala sa više mrežnih adresa (standardno koristi bilo koju adresu),
- `notify-source` - određuje koja će izvorišna adresa biti korištena za slanje NOTIFY poruka; ovo je prilično važan parametar budući da mora odgovarati adresi za koju primarni poslužitelj očekuje i koju zna iz odgovarajućih NS zapisa (standardno koristi bilo koju adresu),
- `provide-ixfr` - određuje da li primarni poslužitelj omogućava inkrementalni prijenos zone ako ga sekundarni zatraži (standardno omogućeno),
- `request-ixfr` - određuje da li će sekundarni poslužitelj tražiti inkrementalni prijenos zone od primarnog (standardno omogućeno).

Primjer 19: Options odjeljak iz `named.conf` datoteke

```
options {  
    directory "/etc/bind";  
    query-source address * port 53;
```

```
allow-transfer { xfer; };
allow-recursion { trusted; };
version "Unknown";
transfer-format many-answers;
max-transfer-time-in 120;
interface-interval 120;
notify yes;
recursion yes;
minimal-responses yes;
notify-source 161.53.116.8;
transfer-source 161.53.116.8;
};
```

3.4.Pristupne liste

Direktiva `acl` služi definiranju pristupnih listi, odnosno definiranju simboličkog imena za grupu adresa koje će se kasnije koristiti u konfiguraciji. Postoji par ugrađenih pristupnih listi koje imaju posebne namjene:

- `any` - odgovara svim adresama,
- `none` - odgovara niti jednoj adresi,
- `localhost` - odgovara svim IPv4 i IPv6 adresama lokalnog poslužitelja,
- `localnets` - odgovara svim lokalnim mrežama u kojima se nalazi poslužitelj, odnosno svim adresama iz takvih mreža.

Sintaksa za ovu naredbu je jednostavna:

```
acl acl-name {
    address_match_list
};
```

U primjeru ćemo definirati nekoliko pristupnih listi koje smo koristili u `options` odjeljku:

Primjer 20: Definiranje pristupnih listi u `named.conf`

```
acl "xfer" {
    161.53.72.21;
    161.53.3.7;
    161.53.2.70;
    127.0.0.1;
};

acl "trusted" {
    161.53.116.0/22;
    193.198.206.0/24;
    193.198.217.192/27;
    localhost;
};
```

3.5. Odjeljak za zapisnike

Rečena `logging` direktiva određuje gdje i kada će biti zapisane poruke o greškama, informativne i ine poruke. Specifično, `channel` dio definira simboličko ime i određuje kakve će biti izlazne metode, oblici ispisa i nivoi. Dotično simboličko ime se kasnije koristi sa `category` parametrom da se odredi kako i gdje će se različiti tipovi poruka zapisivati. Sintaksa je sljedeća:

```
logging {
  [ channel channel_name {
    ( file path name
      [ versions ( number | unlimited ) ]
      [ size size spec ]
      | syslog syslog_facility
      | stderr
      | null );
    [ severity (critical | error | warning | notice |
               info | debug [ level ] | dynamic ); ]
    [ print-category yes or no; ]
    [ print-severity yes or no; ]
    [ print-time yes or no; ]
  }; ]
  [ category category_name {
    channel_name ; [ channel_name ; ... ]
  }; ]
  ...
};
```

Standardno sav ispis se usmjerava na jedan ili više kanala (definiranih `channel` parametrom). Postoji nekoliko predefiniranih standardnih kanala:

- `default_syslog` - šalje `syslog` programu sa `daemon` oznakom, te šalje samo srednje i jako kritične informacije (informacije i više),
- `default_debug` - sprema u datoteku `named.run` u tekućem direktoriju sve poruke koje generira servis,
- `default_stderr` - ispisiše sve greške na standardni izlaz za greške,
- `null` - sve što se ovdje zaprimi se nigdje ne ispisiše.

Pomoću kategorija (definiranih `category` parametrom) se definira gdje će koja kategorija poruka i kako završiti. Opet, postoji niz standardnih kategorija koje definiraju tipove poruka koje generira servis:

- `default` - općenite postavke za sve poruke,
- `general` - sve inače neklasificirane poruke,
- `unmatched` - poruke za koje nije moguće bilo odrediti klasu/tip,
- `database` - poruke vezane uz baze za spremanje zona i međuspremnik,
- `security` - poruke vezane uz sigurnost, odbijanje zahtjeva klijentima i sl,
- `config` - vezano uz čitanje i obradu konfiguracijskih datoteka,
- `resolver` - vezano uz DNS rezoluciju, rekurzivne upite i sl,

- `xfer-in` i `xfer-out` - poruke za prijenos zona,
- `notify` - informacije o NOTIFY porukama,
- `client` - obrada zahtjeva klijenata,
- `network` - poruke vezane uz mrežna sučelja i komunikaciju,
- `update` i `update-security` - DDNS i sigurnosne poruke vezane uz isti,
- `queries` - detaljne informacije o upitima od klijenata,
- `dispatch` - informacije o razdiobi paketa unutar samog servisa,
- `dnssec` - DNSSEC i TSIG informacije,
- `lame-servers` i `delegation-only` - problemi u delegaciji i loše konfiguracije.

U praksi je većina postavki standardno dobro postavljena i koriste se standardni sistemski programi za spremanje zapisnika (koristi se `syslog` program i `daemon` kao oznaka servisa), ali je neke nepotrebne kategorije zgodno eliminirati, kao u primjeru:

Primjer 21: Korištenje logging direktive

```
logging {  
    category lame-servers { null; };  
};
```

3.6. Odjeljak kontrole

Direktiva `controls` određuje kanale za upravljanje DNS servisom, no za sada takve kanale jedino koristi `rndc` program koji je inače dio Bind distribucije. Sintaksa je sljedeća:

```
controls {  
    inet ( ip_addr | * ) [ port ip_port ] allow  
    { address_match_list }  
      keys { key_list };  
    [ inet ...; ]  
};
```

Kontrolni `inet` kanal je TCP port na željenoj adresi koji sluša nadolazeće naredbe sa odgovarajućih računala i izvršava ih. Preporučljivo je dozvoliti isključivo lokalno računalo (127.0.0.1) za upravljanje, da se smanji mogućnost zlouporabe. Osnovni model autorizacije je (uz postojeću listu dozvoljenih adresa) formiran ključevima, koji moraju odgovarati kod poslužitelja i kod klijenta. U slučaju da nema definiranog `controls` odjeljka, servis će postaviti standardni kanal koji sluša na 127.0.0.1 i ::1 adresi i portu 953, što je i preporučljivo. Sam ključ će pokušati iščitati iz datoteke `rndc.key` tražeći ga u direktoriju određenim `directory` parametrom u `options` odjeljku ili u `/etc`. Dotičnu datoteku moguće je automatizirano napraviti koristeći sljedeću naredbu, a time definiramo ključ istovremeno i za `rndc` i za servis, budući da je oba čitaju:

```
rndc-confgen -a
```

Primjer 22: Controls odjeljak iz named.conf datoteke

```
controls {  
    inet 127.0.0.1 allow { localhost; };  
    inet * port 9999 allow { "rndc-remote-users"; } keys  
    { "rndc-remote-key"; };  
};
```

3.7. Odjeljak ključeva

Dotični `key` odjeljak definira dijeljene autorizacijske ključeve za TSIG ali i za komunikacijske kanale za upravljanje servisom. Svaki ključ ima svoje simboličko ime `key_id`, algoritam `algorithm` (isključivo `hmac-md5`) i niz znakova `secret` koji je zapravo ključ u formi base-64 kodiranog niza znakova. Sintaksa je sljedeća:

```
key key_id {  
    algorithm string;  
    secret string;  
};
```

Primjer 23: Ključ za rndc program i za Bind servis

```
key "rndc-remote-key" {  
    algorithm hmac-md5;  
    secret "OmItW1lOyLVUEuvv+Fme+Q==";  
};
```

3.8. Server odjeljak

Osnovna namjena `server` odjeljka je definirati karakteristike poslužitelja u interakciji sa drugim poslužiteljima, koje specifično imenujemo IP adresom. Sintaksa je sljedeća:

```
server ip_addr {  
    [ bogus yes_or_no ; ]  
    [ provide-ixfr yes_or_no ; ]  
    [ request-ixfr yes_or_no ; ]  
    [ edns yes_or_no ; ]  
    [ transfers number ; ]  
    [ transfer-format ( one-answer | many-answers ) ; ]  
    [ keys { string ; [ string ; [...]] } ; ]  
    [ transfer-source (ip4_addr | *) [port ip_port] ; ]  
    [ transfer-source-v6 (ip6_addr | *) [port  
ip_port] ; ]  
};
```

A značenje parametara je redom (spominjemo samo nove parametre, jer je dio već obrađen u `options` odjeljku):

- `bogus` - označava da će udaljeni poslužitelj za kojeg se otkrije da daje neispravne DNS podatke biti označen nevaljanim, te mu budući upiti više neće biti davani (standardno nije omogućeno),
- `edns` - definira omogućene EDNS0 ekstenzije (standardno omogućeno),
- `transfers` - definira broj paralelnih ulaznih prijenosa zona od pojedinačnog poslužitelja,
- `keys` - omogućava korištenje predefiniranih ključeva iz `key` odjeljka, a dotični se koriste za sigurnosne DNS transakcije.

Primjer 24: Korištenje server odjeljka

```
server 161.53.3.7 {  
    bogus yes;  
};
```

3.9. Odjeljak konfiguracije pogleda

Vjerojatno jedan on najkorisnijih dijelova konfiguracije je `view`. Dotični tip omogućava konfiguriranje DNS poslužitelja na takav način da se serviraju različite informacije u ovisnosti o adresi klijenta. Svaki pojedinačni ovakav odjeljak definira jedan pogled koji se servira klijentima koji odgovaraju potparametru `address_match_list` iz `match-clients` i `match-destinations` parametara. Klijente je moguće određivati i pomoću ključeva odnosno `keys` parametara, ali i pomoću tipa upita, recimo `match-recursive-only` će promijeniti pogled samo na rekurzivnim upitima. U cijeloj definiciji pogleda je iznimno bitan redoslijed, budući da on definira koja će se akcija prva odvit. Dobar dio parametara iz `options` odjeljka se također može specificirati za pojedini pogled. Ako je pojedina zona definirana unutar nekog `view` odjeljka, ona će biti isključivo dostupna klijentima koji odgovaraju tom odjeljku - pa je na taj moguće imati više zona sa istim imenom ali u različitim pogledima, što ostvaruje princip razdijeljenog DNS poslužitelja. Moguće je i ne koristiti ovakve odjeljke, međutim tada je implicitno definiran interni pogled u kojem se automatski nalaze sve globalno definirane zone i svi parametri postavljeni u `options` odjeljku. U protivnom, zone ne smiju biti globalno definirane već isključivo unutar `view` odjeljaka. Sintaksa je sljedeća:

```
view view_name  
    [class] {  
        match-clients { address_match_list } ;  
        match-destinations { address_match_list } ;  
        match-recursive-only yes_or_no ;  
        [ view_option; ...]  
        [ zone_statement; ...]  
    };
```

Kao što smo već rekli, svaki pogled definiran svojim simboličkim imenom utiče nasamo na klijente koji mu odgovaraju kroz jedan od tri moguća načina (dovoljno je da odgovara bilo koji od njih): `match-clients` (izvorišne adrese klijenata), `match-destinations` (odredišne adrese) ili `match-recursive-only` (samo rekurzivni upiti).

Primjer 25: Razdijeljeni DNS kroz view direktive

```
view "internal" {
    // interna mreža
    match-clients { 10.0.0.0/8; };
    // pružamo rekurziju
    recursion yes;
    // kompletan pogled na zonu
    zone "example.com" {
        type master;
        file "example-internal.db";
    };
};

view "external" {
    // svi klijenti koji nisu odgovarali gornjem bloku
    // (bitan je redoslijed blokova!)
    match-clients { any; };
    // vanjski klijenti nemaju prava rekurzije
    recursion no;
    // pružamo samo željene vanjske adrese
    zone "example.com" {
        type master;
        file "example-external.db";
    };
};
```

3.10. Umetnuta konfiguracijska datoteka

Rečena `include` direktiva omogućava umetanje dodatne datoteke u konfiguraciju (`named.conf`) točno na mjestu gdje je `include` direktiva. Na ovaj način je složene konfiguracije moguće logičnije razdijeliti, no obično samo dovodi do konfuzije. Prava primjena je sa slučajeve autogeneriranih dijelova konfiguracije, gdje može biti statički osnovni dio konfiguracije koji učitava daljnje dijelove. Trivijalna je sintaksa:

```
include filename;
```

Dodatne konfiguracijske datoteke se koriste primjerice za RFC 1918 domene ili recimo automatski dohvaćene "nepoćudne" domene:

Primjer 26: Umetnute konfiguracijske datoteke

```
include "/etc/bind/zones.rfc1918";
```

```
include "/etc/bind/spywaredomains.zones";
```

3.11. Odjeljak za zone

Rečeni odjeljak `zone` je u najvažniji odjeljak za sve autoritativne DNS poslužitelje. Kroz iste odjeljke se definiraju sve osobine i funkcionalnosti za pojedinu zonu koja se poslužuje, sa mogućnošću redefiniranja svih globalno postavljenih parametara. Sintaksa je sljedeća:

```
zone zone_name [class] [{
    type ( master | slave | hint | stub | forward |
delegation-only ) ;
    [ allow-notify { address_match_list } ; ]
    [ allow-query { address_match_list } ; ]
    [ allow-transfer { address_match_list } ; ]
    [ allow-update { address_match_list } ; ]
    [ update-policy { update_policy_rule [...] } ; ]
    [ allow-update-forwarding { address_match_list } ; ]
    [ also-notify { ip_addr [port ip_port] ; [ ip_addr
[port ip_port] ; ... ] } ; ]
    [ check-names (warn|fail|ignore) ; ]
    [ dialup dialup_option ; ]
    [ delegation-only yes_or_no ; ]
    [ file string ; ]
    [ forward (only|first) ; ]
    [ forwarders { ip_addr [port ip_port] ; [ ip_addr
[port ip_port] ; ... ] } ; ]
    [ ixfr-base string ; ]
    [ ixfr-tmp-file string ; ]
    [ maintain-ixfr-base yes_or_no ; ]
    [ masters [port ip_port] { ( masters_list | ip_addr
[port ip_port] [key key] ) ; [...] } ; ]
    [ max-ixfr-log-size number ; ]
    [ max-transfer-idle-in number ; ]
    [ max-transfer-idle-out number ; ]
    [ max-transfer-time-in number ; ]
    [ max-transfer-time-out number ; ]
    [ notify yes_or_no | explicit ; ]
    [ pubkey number number number string ; ]
    [ transfer-source (ip4_addr | *) [port ip_port] ; ]
    [ transfer-source-v6 (ip6_addr | *) [port
ip_port] ; ]
    [ alt-transfer-source (ip4_addr | *) [port ip_port] ;
]
    [ alt-transfer-source-v6 (ip6_addr | *) [port
ip_port] ; ]
    [ use-alt-transfer-source yes_or_no ; ]
    [ notify-source (ip4_addr | *) [port ip_port] ; ]
    [ notify-source-v6 (ip6_addr | *) [port ip_port] ; ]
    [ zone-statistics yes_or_no ; ]
```

```
[ sig-validity-interval number ; ]  
[ database string ; ]  
[ min-refresh-time number ; ]  
[ max-refresh-time number ; ]  
[ min-retry-time number ; ]  
[ max-retry-time number ; ]  
[ multi-master yes_or_no ; ]  
[ key-directory path_name ; ]  
  
}];
```

Svaka zona ima svoj tip `type` koji određuje način prihvata i posluživanja domenskih informacija:

- `master` - poslužitelj ima osnovnu, glavnu kopiju podataka za zonu i treba biti autoritativni primarni poslužitelj,
- `slave` - poslužitelj treba biti autoritativni sekundarni poslužitelj za danu zonu. Ovaj tip poslužitelja mora nužno imati i postavljenu `masters` listu u kojoj se nalazi jedna ili više IP adresa primarnih poslužitelja sa kojih sekundarni može kopirati (AXFR ili IXFR) zonu. Standardno se preporuča i korištenje datoteke kroz `file` parametar, čime je moguće definirati mjesto gdje se zapisuje cijela zona pri uspješnom prijenosu i time se ubrzava ponovno podizanje poslužitelja i smanjuje broj nužnih prijenosa zone. Na poslužiteljima sa vrlo velikim brojem zona se preporuča izbjegavati nakupljanje većeg broja datoteka u istom direktoriju, pa ih je zgodnije rasporediti po nizu poddirektorija,
- `stub` - poseban tip poslužitelja specifičan za Bind. Riječ je o vrsti sekundarnog poslužitelja koji od primarnog prihvaća i poslužuje isključivo NS zapise. Primarna funkcija takvog načina je eliminacija povezujućih zapisa, ali danas se vrlo rijetko sreće u praksi,
- `forward` - služi definiranju prosljeđivanja DNS upita na razini pojedine zone. Da bi se zona mogla prosljeđivati, nužno je imati već spomenute `forward` i/ili `forwarders` parametre u dotičnom odjeljku,
- `hint` - zona koja definira inicijalnu grupu korijenskih DNS poslužitelja. DNS softver pri pokretanju koristi tu grupu da bi kontaktirao barem jedan poslužitelj i saznao aktualni popis. U slučaju da ovakva zona nije definirana, Bind će koristiti internu i potencijalno zastarjelu listu,
- `delegation-only` - također poseban tip zone koji služi prisilnoj delegaciji. Svaki odgovor koji ne sadrži implicitnu ili eksplicitnu delegaciju u odjeljku autoriteta će se tretirati kao greška NXDOMAIN. Ovakav tip zapisa se uglavnom koristi isključivo kod TLD zona, a nikad kod poddomena.

Što se tiče klasa, one mogu biti HS (Hesiod), CH (Chaos) i IN (Internet). Standardno se klasa `class` ne piše, a podrazumijeva se IN tip. Što se pak tiče opcija, one su uglavnom iste iz `options` odjeljka, te je moguće u potpunosti redefinirati željeno ponašanje za pojedinu zonu.

3.12. Konfiguracija zona

Sada kada su razjašnjeni svi odjeljci `named.conf` konfiguracije, ostaje još pokazati konfiguraciju samih **zonskih datoteka** koje sadrže odgovarajuće RR koji se poslužuju. Par osnovnih pravila za formiranje ispravne zonske datoteke:

- Zona se sastoji od komentara, parametara i RR-ova,
- Komentari isključivo počinju sa ";" znakom i protežu se do kraja reda. Niti jedan drugi tip komentara nije podržan, te umetanje krivog znaka može uzrokovati odbacivanje većeg dijela datoteke,
- Parametri započinju isključivo za znakom "\$". Riječ je o `$ORIGIN`, `$INCLUDE`, `$TTL` i nestandardnom (i relativno kompliciranom) `$GENERATE`.
- Svaki RR mora biti u odgovarajućem formatu:
`ime [ttl] [klasa] rr podaci-specifični-za-rr,`
- `$TTL` bi trebao uvijek biti prisutan kao prvi RR u datoteci,
- Prvi RR (ne računajući `$TTL`) mora biti SOA.

Vrijedi još nekoliko pravila koje je vrijedno upamtiti:

- **Pravilo više zapisa:** ako se više slijednih zapisa odnosi na istu labelu, dovoljno je navesti ime za prvi RR, a ostali ime mogu imati prazno. Općenito, zapisi za istu labelu ne moraju nužno biti jedan za drugim - ali se onda uvijek mora pisati ime labele. Zapisi sa istom labelom će biti posluživani cikličkim redoslijedom,
- **Znak promjene značenja:** znak "\" se koristi za onemogućenje specijalnog značenja za pojedini znak, te se fizički postavlja ispred željenog znaka. Kako se recimo " znak koristi za određivanje niza znakova, tako je unutar postojećeg niza znakova nužno koristiti znak promjene značenja ako želimo imati i jedan ili više navodnika unutar istog niza,
- **Prazni znakovi** se ignoriraju: tab znakovi i razmaci se ignoriraju. Moguće ih je slobodno koristiti radi poboljšanja čitljivosti zonskih datoteka,
- **Velika i mala slova:** u zonama se ne razlikuju velika i mala slova,
- **Točka na kraju labele:** Ako je točka na kraju imena u RR ili parametru, onda je ime ispravno određeno - te ako sadrži potpuno ime uključujući i ime računala, onda je FQDN. Vrijednost imena će se upotrebljavati takva kakva je, bez promjena. Ako nema točke na kraju imena, onda ime nije ispravno određeno te će DNS softver automatski dodati ime zone (iz odgovarajućeg `zone` odjeljka ili iz `$ORIGIN` parametra) na kraj svake takve labele.

Rečeni parametri imaju sljedeća značenja:

- **\$INCLUDE** - služi umetanju definirane datoteke na mjestu gdje se pojavljuje isti parametar. Sintaksa je sljedeća:

```
$INCLUDE ime_datoteke [ izvorišna_domena ]  
[ komentar ]
```

- **\$ORIGIN** - definira osnovno ime (labelu) koja će se sufiksirati svim nepotpunim labelama (svim imenima koje nisu FQDN odnosno koje ne završavaju točkom). Sintaksa je sljedeća:

```
$ORIGIN ime-domene [ komentar ]
```

- **\$TTL** - definira osnovni TTL za sve zapise koje nemaju specifično definirani TTL. Sintaksa je:

```
$TTL vrijeme [ komentar ]
```

Primjer 27: Korištenje parametara unutar zonskih datoteka

```
$TTL 1D
$ORIGIN primjer.domena.
@ SOA ...
$INCLUDE datoteka.zona
$GENERATE 11-254 $ PTR dhcp$.primjer.domena.
```

U zonama se pojavljuje i **specijalni znak** - u zonskim datotekama znak "@" ima specijalno značenje: gdje se on pojavljuje se smatra da se nalazi ime zone iz odgovarajućeg zone odjeljka. U praksi on ima vrijednost \$ORIGIN.

Što se tiče samih RR-ova, teorijski smo već obradili njihova značenja i uporabu. U zonskim datotekama je moguće koristiti slijedeće RR-ove (spominjemo one najvažnije):

- **A** - IPv4 adresa:

```
;ime          ttl   klasa rr ip
www.carnet.hr.          A   161.53.160.25
esa.fer.hr.    59982 IN    A   161.53.71.180
```

- **AAAA** - IPv6 adresa:

```
;ime          ttl klasa rr ipv6
www.carnet.hr. AAAA
2001:B68:E160:0:20B:DBFF:FEE6:A4F0
```

AAAA zapisi se slobodno mogu miješati sa A zapisima. Uporaba A6 zapisa se još uvijek generalno ne preporuča zbog eksperimentalne naravi.

- **CNAME** - kanoničko, zamjensko ime:

```
;ime          ttl klasa rr      kanoničko_ime
kreator.esa.fer.hr.      CNAME esa1.esa.fer.hr.
www                      CNAME esa1
```

CNAME zapisi imaju praktičnu manu unošenja jednog ili više nužnih dodatnih upita (pažnja, dakle smanjuje se efikasnost pretraživanja DNS

sustava!) da bi se saznala tražena IP adresa iz simboličkog naziva. Dozvoljeno je pokazivati sa jednim CNAME na drugi CNAME, iako je to loša praksa i preporučljivo je to izbjegavati. CNAME RR ne smije dijeliti ime (ovo se odnosi i na praznu labelu, odnosno domenu) niti sa jednim drugim RR-om. Također bi trebalo izbjegavati uporabu CNAME zapisa u NS i MX zapisima zbog mogućih grešaka i komplikacija. CNAME se uglavnom jednostavno u konfiguraciji zamjenjuje sa A zapisom, pa je isti tip zapisa uglavnom preporučljivo upotrebljavati tek kad je to nužno. Još jednom, generalno je pravilo da u ispravnoj zoni ne treba biti CNAME koji pokazuje na drugi CNAME.

- **MX** - definiranje imena i prioriteta SMTP poslužitelja:

```
;ime      ttl klasa rr težina ime
@          MX 4      esa2
esa.fer.hr. MX 5      esa1.esa.fer.hr.
esa.fer.hr. MX 10     hpe50.esa.fer.hr.
```

MX težina je relativno definirana prema težinama ostalih MX RR-ova. Niže vrijednosti se preferiraju, iako je odluka do klijenata (primjerice SMTP poslužitelji). Za svaki MX unutar domene je nužan i odgovarajući A zapis, te upotreba CNAME se nikako ne preporučuje. Dapače, zbog mogućnosti da CNAME može pokazivati na potencijalno netočan zapis (CNAME - A lanci) dio MTA odnosno SMTP poslužitelja eksplicitno odbacuje iste zapise. Stoga možemo zaključiti da MX zapis nikada ne bi smio pokazivati na CNAME.

Napomena: u slučaju da je definirano više MX zapisa za isto ime s istom težinom, dešava se interesantna posljedica. Naime, DNS poslužitelj će ciklički (ili slučajnim odabirom) promijeniti redosljed RR-ova, no to će u većini slučajeva učiniti i sam SMTP poslužitelj interno - nerijetko u suprotnom smjeru od samog DNS poslužitelja. Stoga se u takvim slučajevima preporuča definirati samo jedan MX poslužitelj s jednom labelom, te zatim za tu labelu definirati višestruke A zapise sa istim imenom a različitim IP adresama. Time se operacija kružnog posluživanja predaje isključivo na odluku DNS poslužitelju.

- **NS** - autoritativni DNS poslužitelji za rečenu zonu:

```
;ime      ttl klasa rr ime
srce.hr.   NS bjesomar.srce.hr.
srce.hr.   NS regoc.srce.hr.
```

Kad je riječ o NS za vlastitu zonu, oni se najčešće postavljaju odmah nakon odgovarajućeg SOA, no mogu biti bilo gdje definirani. Preporučljivo je imati barem dva NS po zoni. Ako dotični NS-ovi ukazuju na zapise unutar domene, nužni su i odgovarajući A zapisi i na roditeljskom DNS poslužitelju i na samom poslužitelju u poddomeni. NS ime može biti FQDN, nepotpuna labela, "@" i prazni niz (tretira se kao i \$ORIGIN).

- **PTR** - pružaju reverzno povezivanje IP adrese sa imenom:

```
;ime ttl klasa rr ime
194 PTR esa1.esa.fer.hr.
69 PTR regoc.srce.hr.
```

IP adresa može biti samo jednom navedena za pojedini PTR. U slučaju da više imena dijeli CNAME, A i AAAA - nažalost samo jedna adresa može ići u odgovarajući PTR. Standardno se prakticira da svi IP-ovi koji imaju definirane A zapise imaju i odgovarajući PTR.

- **SOA** - definiranje autoriteta za domen:

```
;ime ttl klasa rr dns-poslužitelj e-mail (
; serijski-broj vrijeme-osvježavanja
; vrijeme-ponovnog-pokušaja vrijeme-isteka
; globalni-TTL )
carnet.hr SOA dns.carnet.hr hostmaster.carnet.hr
( 2005082602 10800 3600 2419200 86400 )
```

U odjeljke za vrijeme je moguće unositi vrijeme i u dmh oblicima (1m i sl.). Otvarajuća zagrada "(" nužno uvijek mora biti u istoj liniji kao i početak SOA zapisa. Postoji isključivo jedan SOA po cijeloj zoni. Smatra se da SOA zapis (vrijedi isto za MX i CNAME zapise) nikad ne bi trebao koristiti CNAME za labelu DNS poslužitelja. Napomena: U slučaju da se nezgodom definira krivi serijski broj (prisjetimo se, općeprihvaćeno je koristiti YYYYMMDDnn oblik serijskog broja) i zone prenesu na sekundarne poslužitelje, više neće biti moguće samo vratiti serijski broj budući da će sekundarni odbacivati prijenos za svaku zonu koja ima manji serijski od njihovog lokalnog. Problem se kod Bind poslužitelja rješava resetiranjem serijskog broja, tako da se na originalni "krivi" serijski broj na primarnom poslužitelju doda vrijednost $2^{31}-1$ te ponovno učitava zona. Nakon toga će se obaviti prijenos takve zone što će resetirati serijski broj na sekundarnim poslužiteljima i zatim je na primarnom moguće upisati željeni serijski broj, koji će funkcionirati na uobičajeni način, a zona će se opet prenijeti dalje.

- **TXT** - definiranje tekstualnog zapisa za računalo ili domen.

```
;ime ttl klasa rr tekst
fsb.hr. TXT "v=spf1 ip4:161.53.116.0/22
ip4:193.198.206.0/24 ip4:193.198.217.192/27 a mx ptr
~all"
srce.hr. TXT "SRCE, Zagreb"
```

- **SRV** - definiranje servisa i poslužitelja za pojedini servis. Riječ je o naprednijoj varijanti MX zapisa koja omogućava definirane proizvoljnih servisa, poslužitelja, težina i prioriteta. Trenutno se koristi intenzivno u Microsoft DC/AD (_udp, _tcp, _msdcs i _sites), OpenLDAP (_ldap) i

različitim VoIP sustavima.

```
;serv.proto.ime ttl klasa rr prio tež port cilj
_http._tcp                SRV 0    1    80    www1
                           SRV 0    3    80    www2

;Microsoft AD (LDAP) primjer
_ldap._tcp.dc._msdcs      SRV 0    0   389    msad
_ldap._tcp                SRV 0    0   389    msad
;zabranimo ostale servise
*._tcp                    SRV 0    0    0     .
*._udp                    SRV 0    0    0     .
```

Za serv polje se koristi IANA simboličko ime servisa (velika i mala slova nisu bitna), specifično za pojedini port. Primijetite da se za svaki servis dodaje znak `_` ispred imena servisa: `_http`, `_ftp`, `_ldap`, itd. Za proto polje se koristi IANA ime protokola, također prefiksirano na isti način: `_tcp`, `_udp`, itd. Polje ime se može izostaviti, pa se automatski dodaje ime domene (`$ORIGIN`), kao što je to i uobičajeno. Polje prio definira prioritet servisa pri čemu je najviši prioritet najniže vrijednosti odnosno 0, dok je najniži prioritet 65535. U slučaju da postoji više servisa sa istim prioritetom, koristi se polje težine - također u rasponu od 0 do 65535, pri čemu 0 definira da se težina ne koristi. U slučaju postojanja više istih SRV RR-ova sa istim prioritetom ali različitim težinama, one definiraju koji će se RR više odnosno češće koristiti u odgovarajućem omjeru težina. Polje port definira koji će se port koristiti za odgovarajući servis - na ovaj se način vrlo jednostavno mogu koristiti proizvoljni portovi za neki servis. Naposljetku, odredište za rečeni servis može i ne mora biti unutar pojedine zone odnosno domene unutar koje se definira sam RR.

4. Djbdns poslužitelj

Djbdns je prilično osebujan softver kojeg je napisao Daniel J. Bernstein kao sigurnu (sa gledišta računalne sigurnosti), brzu i pouzdanu zamjenu za Bind softver koji se u to vrijeme pokazao izrazito problematičnim (desetine sigurnosnih rupa). Riječ je o jednom od nekoliko najpopularnijih DNS poslužitelja u svijetu, a godinama se nije mijenjao zbog svojeg vrlo kvalitetno napisanog koda. Rečeni poslužitelj je zaista izrazito malih zahtjeva za računalnim resursima, a uspješno poslužuje mreže svih veličina i u produkciji pokazuje dobro skaliranje sa opterećenjem.

Arhitekturno se Djbdns razlikuje od Binda prvenstveno po podjeli na niz malih poslužitelja (što je u duhu Unix filozofije) od kojih je svaki zadužen za jedan segment rada:

- **Dnscache** je lokalni DNS poslužitelj i međuspremnik koji ne poslužuje podatke o "vlastitoj" domeni, već isključivo klijentima poslužuje podatke od drugih DNS poslužitelja ili iz vlastitog međuspremnika (komunikacija se obavlja kroz UDP i TCP),
- **Tinydns** je autoritativni DNS poslužitelj koji poslužuje podatke iz vlastite centralizirane baze (ne odgovara na rekurzivne upite, niti na TCP upite),
- **Axfrdns** je poslužitelj za prijenos DNS zona (AXFR, funkcionira isključivo kroz TCP),
- **Walldns** je reverzni DNS vatrozid koji odgovara na iterativne reverzne DNS upite sa "generičkim" odgovorima, sakrivajući na taj način stvarne informacije,
- **RblDNS** je DNS poslužitelj za popise IP adresa, najčešće DNS crne liste e-mail spammera i sličnih.

Očigledno je na prvi pogled da se funkcionalnosti koje sadrži samo jedan centralni proces kod Bind poslužitelja ovdje nalaze u nizu odvojenih poslužitelja. Tako je moguće postaviti minimalnu instalaciju u kojoj se nalaze aktivni samo oni potrebni poslužitelji: npr. za DNS namijenjen samo lokalnim klijentima je dovoljan samo `dnscache`, na nekom drugom računalu može biti samo `tinydns`, DNS poslužitelj koji svijet opslužuje DNS podacima o pojedinoj domeni ili skupu domena. Treće računalo može imati samo `axfrdns`, servis za prijenos zona, itd. Ovakva podjela je dovela do prilično jednostavnije implementacije, manjeg broja grešaka (za sada nije utvrđeno postojanje niti jednog sigurnosnog problema u djbdns servisima) i moćnijeg upravljanja ukupnim mogućnosti koje skup DNS servisa podržava.

Ova raspodjela uloga ima i jednu nezgodnu popratnu pojavu - a to je da se istovremeno `dnscache` DNS međuspremnik i `tinydns` DNS autoritativni poslužitelj ne mogu nalaziti na istoj IP adresi, budući da oba koriste UDP/53 (`axfrdns` servis koristi samo TCP/53). Dakle, u slučaju migracije sa Bind poslužitelja nužni su veći zahvati: autoritativni DNS poslužitelj više ne može biti na istoj IP adresi kao i rekurzivni DNS poslužitelj za klijente. Trivijalno rješenje je obično definirati više IP adresa po jednom mrežnom sučelju i

pridijeliti pojedinom sučelju pojedini servis, pri čemu se obično prakticira `axfrdns` i `tinydns` na jednom sučelju, a `dnscache` na drugom.

Nastavak ovog poglavlja podrazumijeva barem osnovna DNS znanja prikupljena u prethodnim poglavljima kao i instalirane **djbdns**, **ucspi-tcp** (potrebno samo ako se koristi `axfrdns`) i **daemontools** pakete. Daemontools je skup alata nužan za rad i upravljanje svih djbdns servisa - međutim objašnjenja o njegovom radu izlaze iz okvira ovog priručnika.

4.1.Dnscache

Servis `dnscache` je vrlo jednostavan DNS međuspremnički program: njegova osnovna namjena je prihvati rekurzivnih DNS upita od klijenata, saznavanje odgovora od udaljenih DNS poslužitelja, odašiljanje odgovora kao i spremanje istih pozitivnih i negativnih rezultata u međuspremnik. Rečeni servis nikada ne vraća autoritativne podatke (AA zastavica nije nikad postavljena) i uvijek vraća samo informacije dobivene od udaljenih autoritativnih DNS poslužitelja. Autoritativne DNS poslužitelje pronalazi već opisanim prolaskom kroz DNS stablo, prateći delegacije sa odgovarajućih čvorova. Dodatna funkcionalnost je mogućnost unošenja ručno definiranih "prečaca" do pojedine domene, čime je moguće implementirati podijeljenu rezoluciju, odnosno podijeljene poglede na zonu.

Rečeni servis se konfigurira kroz `dnscache-conf` program sa sljedećom sintaksom:

```
dnscache-conf acct logacct D ip
```

Pri tome definiramo direktorij `D` (a to je najčešće `/etc/dnscache`) u kojem će biti konfiguracija servisa. Servis će se pri svakom pokretanju **chrootati** (promijeniti pokazivač osnovnog direktorija - dakle isključivo će moći koristiti datoteke samo iz definiranog direktorija i njegovih poddirektorija) u konfigurirani direktorij i pročitati konfiguraciju u kojoj je definirano da će koristiti korisnika (sa odgovarajućim sistemski definiranim UID-om i GID-om) `acct` za rad, korisnika `logacct` za spremanje sistemskih zapisa u direktorij `direktorij/log/main`. Spremanje sistemskih zapisnika se odvija kroz servis `multilog`, koji garantira i redovno "rotiranje" zapisnika - najinteresantnija je obično tekstualna datoteka `current` koja sadrži tekuće, recentne zapise.

Pri podizanju će rečeni servis započeti osluškivanje na IP adresi `ip` (najčešće `127.0.0.1`). U osnovnoj konfiguraciji se stvara i datoteka `direktorij/root/ip/127.0.0.1`, koja definira da će `dnscache` servis dozvoljavati upite sa adrese `127.0.0.1`. Po potrebi u `root/ip` direktoriju trebate stvoriti prazne datoteke (naredbom `touch`), formirajući time **pristupnu listu**: primjerice ako vam je lokalna mreža `161.53.2.0/24`, potrebno je stvoriti datoteke `161.53.2` i `127.0.0`, čime definirate pristup mreži `161.53.2.0/24` i `127.0.0.0/24`.

Osim `root/ip` direktorija postoji i `root/servers` direktorij: on sadrži liste **autoritativnih poslužitelja** za pojedinu domenu (po jedan poslužitelj u svakom redu datoteke), s time da je ime svake datoteke domena. Datoteka imena `@` sadrži listu 13 vršnih DNS poslužitelja te je nužna za rad poslužitelja. Naravno, u ovom direktoriju možete raditi i već spomenute prečice za prolazak kroz DNS stablo - direktno popisujući poslužitelje za pojedinu domenu.

Primjer 28: Kratice za `dnscache`

Datoteka `srce.hr`:

161.53.2.69

161.53.2.70

Datoteka `2.53.161.in-addr.arpa`:

161.53.2.69

161.53.2.70

Ostatak konfiguracije se nalazi u `env` direktoriju:

- `CACHESIZE` - definira veličinu fiksne tablice međuspremnika, pri čemu se prosječno 5% tablice koristi kao indeks tablice. Tablica se nikad ne smanjuje niti ne povećava: u slučaju "preljeva" se najstariji zapis briše i tako kružno. Dotična tablica se rezervira pri pokretanju samog procesa,
- `DATALIMIT` - služi kao zaštita od slučajne greške u samom servisu. Rečena varijabla će postaviti maksimalnu veličinu do koje će servis smjeti rasti - a praksa je obično postaviti vrijednost na par MB veće od `CACHESIZE` varijable,
- `IP` - definira IP adresu na kojoj sluša sam DNS servis. Standardno je dozvoljena svega jedna adresa, tipično 127.0.0.1 za jedno računalo, ili IP adresa nekog fizičkog Ethernet sučelja za opsluživanje više računala,
- `IPSEND` - definira IP adresu sa koje DNS servis šalje DNS odgovore i upite. Tipično je to 0.0.0.0, što znači da će koristiti adresu prvog mrežnog sučelja na sustavu. No, takvo nešto je najčešće nepoželjno u sustavima sa više IP adresa (a klijenti smiju paranoično odbacivati pakete koji su odgovor na komunikaciju ali dolaze sa krive IP adrese), pa se preporuča koristiti adresu navedenu u `IP` varijabli,
- `ROOT` - direktorij u kojem se nalazi cijela konfiguracija servisa i svi navedeni poddirektoriji,
- `FORWARDONLY` - u slučaju postojanja ove datoteke će servis koristiti datoteku `servers/@` kao listu IP adresa udaljenih DNS poslužitelja kojima će prosljeđivati sve upite, dok sam neće obavljati nikakvu rezoluciju,
- `HIDETTL` - u slučaju postojanja ove datoteke se sakriva TTL nad zapisima u odgovorima, te će svi uvijek iznositi 0.

Spomenimo još par specifičnosti za `dnscache` servis:

- Nikad se ne šalje odgovor sa AA zastavicom postavljenom,

- DNS odgovori nikad ne sadrže odjeljak autoriteta ili dodatni odjeljak, a u slučaju da je DNS odgovor veći od dozvoljenog - on se u potpunosti odbacuje,
- Uvijek se odbacuju sljedeći upiti: prijenos zone, nerekurzivni (iterativni) upiti i inverzni upiti (IQUERY),
- DNS A upit za `localhost` se interno uvijek obrađuje i odgovara IP adresi `127.0.0.1`. Isto vrijedi i za PTR `1.0.0.127.in-addr.arpa`, koji odgovara labeli `localhost`,
- Za rad servisa je nužno postojanje `@` datoteke, te ne postoji ugrađena lista vršnih poslužitelja kao kod Bind servisa,
- Dodatni zapisi u DNS odgovorima koji nisu dio domene za koju je NS autoritativan se ignoriraju (niti se ne vraćaju klijentima, niti se stavljaju u međuspremnik) radi sigurnosnih razloga,
- Povezujući zapisi se tretiraju vrlo oprezno - povezivanje nije dozvoljeno izvan domene za koju je NS autoritativan, nije dozvoljen TTL 0 u povezujućim zapisima niti je dozvoljeno ikakvo povezivanje koje krši DNS RFC-ove,
- Zapisi u međuspremniku vrijede maksimalno jedan tjedan, dok zapisi sa TTL od 2147483647 bivaju tretirani kao TTL 0,
- SOA zapisi se nikad ne spremaju u DNS međuspremnik,
- Standardno se paralelno obrađuje maksimalno 200 istovremenih UDP upita i 20 istovremenih TCP upita (ovo je moguće promijeniti u izvornom kodu) - novopristigli upit iznad granice paralelizma rezultira sa odbacivanjem najstarijeg,
- Standardno se sluša samo na jednoj IP adresi (ovo je moguće promijeniti u izvornom kodu).

4.2. *Tinydns*

Servis `tinydns` je minimalni autoritativni DNS poslužitelj. On zaprima iterativne DNS upite iz lokalne mreže i svijeta te odgovara sa DNS odgovorima iz unaprijed definirane baze. Odgovor na upit koji se ne može naći u bazi se jednostavno ne odgovara. Jedino u slučaju da se u bazi pronade zapis koji definira autoritet nad domenom, onda se na upit za nepostojećim zapisom u dotičnoj domeni odgovara sa `NXDOMAIN` odgovorom. Standardno se upiti serviraju iz `data.cdb` baze koja je u specifičnom internom formatu. Rečena baza na disku je binarna datoteka obično male veličine i relativno dobrih performansi; svi upiti se rješavaju u svega dva pristupa bazi, budući da je upit ključ a podatak odgovor. Dotična baza se koristi i za `axfrdns` servis, koji iz nje čita podatke potrebne za prijenos zone.

Slično kao i `dnscache` servisu, `tinydns` servis se konfigurira kroz `tinydns-conf` program sa sljedećom sintaksom:

```
tinydns-conf acct logacct D ip
```

Pri tome definiramo direktorij `D (/etc/tinydns` u većini slučajeva) u kojem će biti konfiguracija servisa. Servis će se pri svakom pokretanju **chrootati** u konfigurirani direktorij (`env/ROOT` datoteka) i pročitati konfiguraciju u kojoj je definirano da će koristiti korisnika `acct` za rad, a korisnika `logacct` za spremanje sistemskih zapisa u direktorij `direktorij/log/main`.

Rečeni servis će raditi na IP adresi `ip` (najčešće je to adresa nekog vanjskog Ethernet sučelja), koje se definira u `env/IP` datoteci kao i kod `dnscache` servisa. Kako je `tinydns` zamišljen kao servis za sve korisnike na Internetu, ne postoji nikakva pristupna lista. Za formiranje različitih pogleda se obično koristi više odvojenih `tinydns` servisa, pri čemu je svaki u vlastitom direktoriju i sa vlastitom **bazom**. Primijetite da `tinydns` odgovara isključivo na jednostavne iterativne UDP/53 upite. Svi drugi tipovi upita se odbacuju, a to su npr: prijenosi zone (AXFR, IXFR), inverzni upiti (IQUERY), upiti za klasama koje nisu IN, nepotpuni paketi, TCP upiti i višestruki upiti.

Osim rečene dvije konfiguracijske datoteke u `env` direktoriju, servis je karakterističan po tome što se većina konfiguracije odvija u `root` direktoriju, za razliku od `dnscache` servisa:

- `Makefile` - `make` skripta za stvaranje binarne `data.cdb` datoteke iz `data` konfiguracije zone koristeći `tinydns-data` program,
- `add-alias` - program za dodavanje **CNAME** u zonu, pri čemu stvara "+" zapis,
- `add-childns` - program za dodavanje delegacije za zonu, pri čemu stvara "&" zapis,
- `add-host` - program za dodavanje **A** i **PTR** zapisa u zonu, što je jedan "=" zapis,
- `add-mx` - program za dodavanje **MX** zapisa u zonu, pri čemu stvara "@" zapis,
- `add-ns` - program za definiranje **NS** za zonu, pri čemu stvara "." zapis,
- `data` - konfiguracija zone u obliku čistog teksta,
- `data.cdb` - binarna datoteka stvorena iz `data` zone koristeći `tinydns-data` program.

Uz osnovni `tinydns-conf` postoje još dva vrlo bitna programa:

- `tinydns-data` - čita konfiguraciju jedne ili više zona iz `data` datoteke i stvara `data.cdb` binarnu datoteku. Cijela operacija se obavlja atomično, pa se smije koristiti i dok je servis aktivan. Nakon svake promjene izvorišne datoteke nužno je svaki put ručno pokrenuti rečeni program. O samom formatu izvorišne datoteke će još biti riječi.
- `tinydns-edit` - omogućava editiranje `data` datoteke, odnosno dodavanje novih zapisa. Za svaku naredbu dodavanja postoji odgovarajuća `add-` skripta, a dotične su objašnjene u gornjem odlomku.

4.3. *Tinydns zone*

Naposljetku, opišimo i konfiguraciju samih DNS podataka, odnosno način pisanja zapisa u `data` datoteci:

- Svaki pojedini DNS zapis je u vlastitom redu,
- Svaki red započinje sa znakom koji definira o kojem je tipu DNS zapisa riječ,
- Unutar cijelog reda postoji nekoliko odjeljaka, koji su odvojeni znakom dvotočke ":",
- Pojedini odjeljak se smije izostaviti, ali ne i dvotočke koje ga okružuju,
- Razmaci, tabulatori i prazni redovi se ignoriraju u datoteci,
- Svaki zapis (red) može imati vlastiti TTL, no on se smije izostaviti pa će se koristiti standardne vrijednosti,
- Svaki zapis može imati i vlastitu **vremensku oznaku** (engl. timestamp) koja se u praksi rijetko koristi, a koja ima dvojno značenje:
 - U slučaju da je TTL različit od 0 ili izostavljen (dakle `tinydns` se brine o njegovoj vrijednosti), to je vrijeme od kojeg će zapis početi vrijediti - odnosno rečeni zapis će se ignorirati do tog vremena,
 - Kad je TTL jednak 0, rečena oznaka predstavlja **TTD** (engl. time to die) odnosno vrijeme prestanka aktualnosti rečenog zapisa.

Tipovi zapisa su sljedeći:

- `.` (točka) - definira NS+A+SOA zapise za domenu `fqdn` zapisanu u FQDN formatu. Sintaksa je sljedeća:

```
.fqdn:ip:x:ttl:timestamp:lo
```

Pri tome će se u zoni pojaviti **NS** zapis `x.ns.fqdn` kao poslužitelj za domenu `fqdn`; **A** zapis za `x.ns.fqdn` sa adresom `ip`; **SOA** zapis za domenu `fqdn` koji ukazuje na `x.ns.fqdn` kao primarni DNS poslužitelj i imat će `hostmaster@fqdn` kao kontakt adresu. U nekim slučajevima je ovakav "magični" tip definiranja višestrukih RR-ova koristan, a u nekim slučajevima se koriste drugi tipovi zapisa o kojima će biti riječi u nastavku. Zapis `lo` je područje djelovanja zapisa, o kojem ćemo na samom kraju nešto više spomenuti. Zapis `ip` je moguće izostaviti. Ako zapis `x` sadrži točku, onda se tretira kao FQDN zapis; dakle neće se magično dodati `ns.fqdn` sufiks na ime poslužitelja.

Primjer upotrebe bi bio sljedeći:

```
.esa.fer.hr:161.53.71.194:esa1.esa.fer.hr
```

Time će se stvoriti NS zapis za `esa.fer.hr` domenu koji definira `esa1.esa.fer.hr` kao DNS poslužitelj za istu. Osim toga, stvara se A zapis koji definira da `esa1` ima adresu `161.53.71.194`. Naposljetku, definira se i već opisani SOA, sa `hostmaster@esa.fer.hr` adresom za domenu `esa.fer.hr`.

- **& (i)** - definira NS+A zapise za domenu fqdn:

```
&fqdn:ip:x:ttl:timestamp:lo
```

Analogno "." zapisu, ovaj će zapis rezultirati stvaranjem **NS** zapisa u zoni za `x.ns.fqdn` kao poslužitelja za domenu `fqdn`. Također, formirati će se i **A** zapis za `x.ns.fqdn` sa adresom `ip`. Naravno, "&" zapisa smije biti nekoliko za zonu, budući da svaki definira individualni DNS poslužitelj za rečenu domenu.

Primjer upotrebe:

```
&71.53.161.in-addr.arpa::esa1.esa.fer.hr.:86400
&71.53.161.in-addr.arpa::hobbit.fsb.hr.:86400
&esa.fer.hr::esa1.esa.fer.hr.:86400
&esa.fer.hr::hobbit.fsb.hr.:86400
&::a.root-servers.net
```

Ovime smo definirali dva DNS poslužitelja `esa1` i `hobbit` za `71.53.161.in-addr.arpa` domenu (dakle reverzni DNS za `161.53.71.0/24` mrežu) sa TTL od 1D. Osim toga smo definirali i dva poslužitelja `esa1` i `hobbit` za `esa.fer.hr` domenu. Primijetite interesantnu činjenicu da je jedan od poslužitelja za domenu izvan same `esa.fer.hr` domene. Posljednji zapis se brine za slučajeve **krive delegacije**, budući da `tinydns` inače ne odgovara na upite koji su izvan njegovog autoriteta (za koje nema SOA zapis) jer se smatra da je to posao `dnscache` servisa. U pojedinim konfiguracijama (sjedimo se, `dnscache` i `tinydns` nisu na istom sučelju) se može desiti da zahtjev za rezolucijom DNS poslužitelja izvan zone bude problem (upit završava na autoritativnom DNS poslužitelju) - koji se onda rješava ovom tehnikom, koja imitira ponašanje Bind poslužitelja.

- **= (jednako)** - definira A+PTR zapise:

```
=fqdn:ip:ttl:timestamp:lo
```

Ovakav tip će stvoriti **A** zapis za `fqdn` labelu sa IP adresom `ip`, te **PTR** zapis za `d.c.b.a.in-addr.arpa` prema labeli `fqdn`, ako je IP adresa u obliku `a.b.c.d`. Naravno, moguće je odvojeno stvoriti **A** i **PTR** kao što se inače radi u Bind zonama, a što ćemo također pokazati u kasnijim odlomcima. Još jednom ponovimo: `tinydns` nikad ne odgovara (uopće ne šalje nikakav odgovor) za zapise za koje nema nadležne "&" ili "." zapise.

Primjer upotrebe:

```
=esa1.esa.fer.hr:161.53.71.194:86400
=hpe50.esa.fer.hr:161.53.71.235:86400
```

- **+** (plus) - definira A zapise:

```
+fqdn:ip:ttl:timestamp:lo
```

Rečeni zapis će stvoriti **A** zapis za labelu `fqdn` sa IP adresom `ip`. U slučaju da postoji više **A** zapisa (stvorenih kroz "+", "=", "@", "." ili "&") oni će se u odgovoru biti po slučajnom (ne ciklički) redoslijedu. U slučaju da ih ima više od 8, u odgovorima će biti slučajni parovi od 8 zapisa.

Primjer upotrebe:

```
+esa1.esa.fer.hr:161.53.71.194:86400
+hpe50.esa.fer.hr:161.53.71.235:86400
```

- **@ (pri)** - definira MX+A zapise:

```
@fqdn:ip:x:dist:ttl:timestamp:lo
```

Kroz ovaj zapis se stvara **MX** zapis za `x.mx.fqdn` u domeni `fqdn` i cijenom (udaljenošću) `dist`. Također se stvara i **A** zapis za `x.mx.fqdn` koji pokazuje na IP adresu `ip`. Ako zapis `x` sadrži točku, onda se tretira kao FQDN zapis; dakle neće se magično dodati `mx.fqdn` sufiks na ime poslužitelja. U slučaju da `dist` nije definiran njegova je podrazumijevana vrijednost 0. Naravno, moguće je stvarati višestruke zapise sa istim ili različitim cijenama.

Primjer upotrebe:

```
@esa.fer.hr::esa1.esa.fer.hr.:5:86400
@esa.fer.hr::hpe50.esa.fer.hr.:10:86400
@esa1.esa.fer.hr::esa1.esa.fer.hr.:5:86400
@esa1.esa.fer.hr::hpe50.esa.fer.hr.:10:86400
```

- **# (ljestve)** - linija komentara. Rečena linija se ignorira u cijelosti.
- **- (minus)** - definira neaktivne zapise:

```
-fqdn:ip:ttl:timestamp:lo
```

Rečeni cijeli zapis će se ignorirati. Obično predstavlja dinamički (kroz kakvo vanjsko sučelje/program za editiranje zona) onemogućena odnosno ugašena računala, iako sam `tinydns` nema mogućnost direktne manipulacije takvim zapisima.

- **' (jednostruki navodnik)** - definira TXT zapise:

```
'fqdn:s:ttl:timestamp:lo
```

Definira **TXT** zapis za labelu `fqdn` sa sadržajem `s`. Moguće je unositi i proizvoljne ASCII znakove koristeći `\NNN` i oktalni zapis unutar niza `s`. Na primjer koristeći `\072` možete definirati dvotočku u nizu znakova, koja bi inače bila tretirana kao razdjelnik.

Primjer za SPF zapis:

```
'esa.fer.hr:v=spf1 ip4\072161.53.71.194
ip4\072161.53.71.235 mx a\072hpe50.esa.fer.hr
a\072esa1.esa.fer.hr a\072esa.fer.hr
mx\072hpe50.esa.fer.hr mx\072esa1.esa.fer.hr
~all:3600
'esa1.esa.fer.hr:v=spf1 a -all:3600
'hpe50.esa.fer.hr:v=spf1 a -all:3600
```

- **^** (kapica) - definira PTR zapise:

```
^fqdn:p:ttl:timestamp:lo
```

Služi definiranju isključivo **PTR** zapisa: stvara se zapis za `fqdn` koji pokazuje na domensko ime `p`. U slučaju korištenja "=" zapisa se stvara **A** i **PTR** istovremeno, a na ovaj način se individualno definira **PTR** i kasnije kroz "+" se po potrebi definira **A** zapis.

Primjer:

```
^194.71.53.161.in-addr.arpa:esa1.esa.fer.hr.:86400
^235.71.53.161.in-addr.arpa:hpe50.esa.fer.hr.:86400
```

- **C** (slovo C) - definira CNAME zapise:

```
Cfqdn:p:ttl:timestamp:lo
```

Stvara kanoničko ime odnosno **CNAME** zapis za `fqdn` koji pokazuje na domensko ime `p`. Kao što smo već napomenuli, nužno je izbjegavati slučajeve u kojima postoji još zapisa za `fqdn`. Kad god koristite **CNAME**, uvijek dobro razmislite da li vam je isti potreban ili možda možete riješiti problem sa standardnim **A** zapisom.

- **Z** (slovo Z) - definira SOA zapise:

```
Zfqdn:mname:rname:ser:ref:ret:exp:min:ttl:timestamp:
lo
```

Stvara samo **SOA** zapis, za razliku od "." zapisa koji stvara još i odgovarajuće **NS** i **A**. Primarni DNS poslužitelj će biti `mname`, `rname` se koristi kao kontakt adresa (s time da se prva točka u labeli pretvara u @, kao i kod Bind poslužitelja - čime dobivamo e-mail adresu), `ser` se tretira kao serijski broj, `ref` kao vrijeme osvježavanja, `ret` kao vrijeme ponovnog pokušaja, `exp` kao vrijeme isteka i `min` kao minimalno dozvoljeno vrijeme. Sva vremena se definiraju u sekundama. Korisna mogućnost je izostavljanje `ser` parametra pri čemu se onda za serijski broj podrazumijeva vrijeme zadnje promjene `data` datoteke - što je zapravo vrlo česta praksa i administratorima pojednostavljuje brigu oko ispravnog serijskog broja. Također je moguće izostaviti `ref`, `ret`, `exp` i

min parametre koji onda respektivno odgovaraju sljedećim vrijednostima: 16384, 2048, 1048576 i 2560.

Primjer:

```
z71.53.161.in-  
addr.arpa:esa1.esa.fer.hr.:postmaster.esa.fer.hr.:2  
8800:7200:604800:604800:86400  
Zesa.fer.hr:esa1.esa.fer.hr.:postmaster.esa.fer.hr.:  
:28800:7200:604800:604800:86400
```

- : (dvotočka) - generički zapisi:

```
:fqdn:n:rdata:ttl:timestamp:lo
```

Riječ je o posebnom tipu zapisa za koji ne postoji ekvivalent u Bind poslužitelju. Naime, koristeći ovaj zapis možete definirati proizvoljan RR kojeg `tinydns` standardno ne podržava. Ovime se formira RR tipa `n` (cijeli broj između 1 i 65535) za labelu `fqdn` sa sadržajem `rdata`. Nužno je izbjegavati definiranje RR-ova koji odgovaraju postojećim zapisima: to su 2 (**NS**), 5 (**CNAME**), 6 (**SOA**), 12 (**PTR**), 15 (**MX**) i 252 (**AXFR**). Način zapisa `rdata` se razlikuje u ovisnosti o tipu zapisa. Kao i za **TXT**, koristeći `\NNN` oktalni zapis je moguće unijeti proizvoljne znakove unutar `rdata`.

Primjer (u komentarima su odgovarajući Bind ekvivalenti):

```
#_http._tcp.esa.fer.hr. 86400 IN SRV 10 100 80  
www.esa.fer.hr  
:_http._tcp.esa.fer.hr:33:\000\012\000\144\000\120\0  
03www\003esa\003fer\002hr\000:86400  
#ipv6-host.esa.fer.hr 86400 IN AAAA  
ffff:1234:5678:9:a:b:c:4321  
:ipv6-  
host.esa.fer.hr:28:\377\377\022\064\126\170\000\011\  
000\012\000\013\000\014\103\041:86400
```

Ostaje nam još pokazati kako funkcionira **područje djelovanja** (engl. client location) zapisa. Područje se definira koristeći "%" (postotak) linije sa sljedećom sintaksom:

```
%lo:ipprefix
```

Pri čemu definiramo da su sve IP adrese koje počinju sa `ipprefix` (moguće ga je izostaviti - onda se smatra da se sve IP adrese podudaraju s istim područjem) u nekom području `lo`. Svako područje `lo` se definira sa jednim ili dva ASCII znaka. Klijent koji traži određeni zapis odgovara samo jednom području - a uspoređivanje se vrši sa što boljim poklapanjem, odnosno poklapanjem u što većem broju bitova. Područje djelovanja se jednostavno dodatno naznačuje za svaki zapis, a njegova pozicija je uvijek na kraju (iza vremenske oznake).

Primjer 29: Područje djelovanja u tinydns zoni

```
%in:10.0
%ex
+esa1.esa.fer.hr:10.0.0.1:86400::in
+esa1.esa.fer.hr:161.53.71.194:86400::ex
```

Postoji i još jedan način unosa podataka u pojedini RR, a to je korištenjem zamjenskih znakova odnosno definiranjem zamjenskih zapisa. Isti su podržani u `*.fqdn` obliku, a kao i kod Bind servisa mijenjaju sve labele osim postojećih. I osim onih definiranih također kroz zamjenske znakove, ali kod kojih postoji veće podudaranje u `fqdn`.

Primjer 30: Zamjenski zapisi u tinydns zoni

```
+www.esa.fer.hr:161.53.71.180:86400
+test.www.esa.fer.hr:161.53.71.235:86400
+*.www.esa.fer.hr:161.53.71.180:86400
```

U gornjem primjeru uspjeli bi upiti za `pero.www.esa.fer.hr`, `jutro.www.esa.fer.hr` i `test.www.esa.fer.hr`. Prva dva vode na 161.53.71.180 IP adresu, dok posljednji vodi na 161.53.71.235 adresu budući da postoji odgovarajući standardni, nezamjenski zapis.

Svi opisani zapisi se standardno nalaze na samo jednom mjestu - unutar već spomenute data datoteke u root direktoriju samog tinydns servisa. Rečena se prevodi u odgovarajući `data.cdb` oblik korištenjem `tinydns-data` naredbe bez argumenata. Alternativa je korištenje naredbe `make` ako postoji na sustavu, koja će opet pozvati gornje naredbe i što je god još potrebno, a definirano je u `Makefile` predlošku.

Postoji još niz nestandardnih dodataka koji dodaju nove zapise (višestruki SOA, NAPTR, SRV, itd), međutim jedino su opisani tipovi oni koje možete očekivati na posve čistoj (kao što je autor zamislio) ili nepoznatoj instalaciji.

4.4.Axfrdns

Rečeni `axfrdns` servis je TCP DNS poslužitelj sa prvenstvenom namjenom prijenosa zone kao odgovora na ispravne **AXFR** upite, ali može služiti i kao autoritativni TCP DNS servis (za DNS odgovore veće od graničnih 512 bajtova). Standardno se u `djbdns` paketu ne preporuča koristiti AXFR za prijenos zona drugom `tinydns` poslužitelju, budući da je za takve operacije nešto bolji eksterni `rsync` alat (koji se koristi npr. kroz `ssh` tunel). Stoga `axfrdns` pokazuje svoju ulogu ponajviše u slučaju kad za primarni poslužitelj koristimo `djbdns`, a za sekundarni Bind ili neki drugi AXFR-kompatibilni poslužitelj.

Napomenimo važan podatak da `tinydns` ne razumije **NOTIFY** mehanizam: on nikad implicitno ne šalje sekundarnim poslužiteljima obavijesti o promjeni zone, već samo eksplicitno odnosno ručno kroz eksterni `tinydns-notify`

alat koji nije dio standardne distribucije. Također ni ne prihvaća NOTIFY, tako da u slučaju kad se na primarnom poslužitelju promijeni zona, sekundarni `tinydns` ne zna za te promjene dokle god se ne izvrši kakva ručna replikacija: na primjer kroz `axfr-get` ili već opisano sa `rsync/ssh` kombinacijom. Sa gledišta suživota sa ostalim servisima, `axfrdns` se najčešće postavlja na isto mrežno sučelje na kojem je i `tinydns`, komplementirajući ga u smislu TCP komunikacije. Dapače, koristi i istu `data` odnosno `data.cdb` datoteku kao izvor podataka o zapisima.

Servis se konfigurira kroz `axfrdns-conf` program sa sljedećom sintaksom:

```
axfrdns-conf acct logacct D tiny ip
```

Servis i njegova konfiguracija će se nalaziti u direktoriju `D` koji je najčešće `/etc/axfrdns`. Servis će se pri svakom pokretanju **chrootati** u konfigurirani direktorij (`env/ROOT` datoteka sa sadržajem parametra `tiny`, odnosno lokacije `tinydns` servisa) i koristiti korisnika `acct` za rad te korisnika `logacct` za spremanje sistemskih zapisa u direktorij `direktorij/log/main`.

Rečeni servis će raditi na IP adresi `ip` (najčešće je to adresa nekog vanjskog Ethernet sučelja), koje se definira u `env/IP` datoteci. Što se pak tiče pristupnih listi, one su definirane u `tcp` datoteci u posebnom formatu o kojem će još biti riječi te su inicijalno zabranjeni svi prijenosi zona (varijabla `AXFR` je prazna). Interesantno je primijetiti da `axfrdns` sam po sebi nema sposobnost komunikacije TCP protokolom već koristi vanjski program `tcpserver` koji mu predaje naredbe preko standardnog ulaza. Rečeni vanjski program je inače dio `ucspi-tcp` paketa od istog autora.

Servis prekida prijenos zone u slučaju greške. To su primjerice:

- Zapunjenost dozvoljene količine memorije,
- Nemogućnost čitanja `data.cdb` datoteke,
- TCP upit veći od 512 bajtova,
- Neispravan upit,
- Nedozvoljen prijenos zone za klijente za koje nije definirana `AXFR` varijabla sa rečenom zonom koja se pokušava prenijeti: rečena varijabla mora sadržavati listu dozvoljenih zona odvojenih znakom `"/` (dijeljeno),
- Upit za RR koji nisu u `data.cdb`.

4.5.Pravila za tcpserver

Kao što smo već spomenuli, `tcpserver` je servis koji omogućava `axfrdns` servisu TCP komunikaciju. Hoće li TCP sjednica biti uspostavljena odlučuju jednostavna pravila u linijskoj tekstualnoj datoteci. To je obično `tcp` datoteka koja se konvertira u binarnu datoteku `tcp.cdb` programom `tcprules`.

Pravila zapisa su sljedeća:

- Svako pravilo je u svojoj vlastitoj liniji,
- Linije koje počinju sa znakom "#" su komentar, te se ignoriraju,
- Svako pravilo je u obliku: `adresa:instrukcije` što definira da će se za TCP vezu sa adrese `adresa` obavljati instrukcije `instrukcije`,
- Linije ne smiju sadržavati razmake, tabove i slične "prazne" znakove,
- Koristiti će se prvo pravilo na koje `tcpserver` naiđe: ne radi se nikakvo složenije podudaranje.

Adrese se zapisuju na sljedeći način:

- `korisnik@ip` - udaljeno računalo odgovara na `auth` upite i `identd` odgovor daje niz `korisnik`,
- `korisnik@=labela` - podrazumijeva se da za IP adresu udaljenog računala postoji i odgovara DNS labela `labela`, te da `identd` odgovor odgovara nizu `korisnik`,
- `=labela` - podrazumijeva se da za IP adresu udaljenog računala postoji dotična DNS labela `labela`,
- Dijelovi IP adrese udaljenog računala koji završavaju sa "." (točkom),
- Dijelovi DNS labele udaljenog računala koji završavaju sa "." (točkom),
- Rasponi IP adresa korištenjem znaka "-" (minus),
- `=` - odgovara bilo kojoj DNS labeli ako ona postoji za IP adresu udaljenog računala,
- Prazan niz - odgovara bilo kojoj IP adresi.

Instrukcije se zapisuju na sljedeći način:

- moraju započeti sa ključnom riječi `allow` (dozvoli vezu) ili `deny` (odbij vezu),
- mogu postavljati neku varijablu okoline u obliku: `var="x"` što će se u našem slučaju koristiti za postavljanje `AXFR` varijable u ovisnosti o adresama klijenata,
- između ključne riječi i postavljanje varijable se koristi znak ",", (zarez).

Primjer 31: `Tcpserver` pravila za `axfrdns`

```
127.0.0.1:allow,AXFR="esa.fer.hr/71.53.161.in-addr.arpa"
161.53.116.8:allow,AXFR="esa.fer.hr/71.53.161.in-addr.arpa"
161.53.2.69:allow,AXFR="esa.fer.hr/71.53.161.in-addr.arpa"
:allow,AXFR=""
```

U našem primjeru dozvoljavamo prijenos dviju zona `esa.fer.hr` i `71.53.161.in-addr.arpa` klijentima sa `127.0.0.1`, kao i klijentima sa `161.53.116.8` i `161.53.2.69` IP adresa. Za sve ostale klijente dozvoljavamo standardne TCP upite osim prijenosa zone tako da postavljamo `AXFR` varijablu na prazan niz.

Naposljetku, datoteka sa pravilima se prevodi u datoteku `tcp.cdb` sa naredbom `tcprules` na sljedeći način:


```
tcprules tcp.cdb tcp.tmp < tcp
```

4.6.Walldns

Servis `walldns` je specifičan i nestandardan DNS servis: njegova jedina dužnost je odgovaranje na iterativne upite iz svijeta o `in-addr.arpa` domenama. Pri tome on odgovara sa umjetnim autoritativnim DNS odgovorima koji služe skrivanju stvarnih informacija o pojedinoj domeni. Za sve IP adrese `x.y.z.w` rečeni će servis posluživati kao da ima definiran slijedeći opći `tinydns` zapis:

```
=w.z.y.x.in-addr.arpa:x.y.z.w
```

Ovime se stvara PTR zapis za neku IP adresu u obliku `x.y.z.w` (odnosno `w.z.y.x.in-addr.arpa`) koji pokazuje na labelu `w.z.y.x.in-addr.arpa` te jedan A zapis za labelu `w.z.y.x.in-addr.arpa` koji pokazuje na IP adresu `x.y.z.w`.

Ovaj servis se u praksi rijetko koristi, pa nećemo objašnjavati njegovo postavljanje.

4.7.Rbldns

Servis `rbldns` je također relativno rijedak slučaj - njegova zadaća je posluživanje RBL podataka. Dakle, rečeni odgovara na iterativne DNS odgovore iz svijeta za A, TXT ili ANY ("*") upitima za IP adresama u obliku `w.z.y.x.base` pri čemu je `base` sufiks (najčešće domena) definiran `BASE` varijablom. Razlog korištenju varijable `BASE` je prvenstveno mogućnost spajanja standardnih `tinydns` podataka i podataka za `rbldns` u jednu datoteku, budući da `rbldns` standardno koristi `data.cdb` za izvor RR odgovora.

Sam servis standardno ignorira inverzne upite, upite koji nisu unutar IN klase, nepotpune pakete, pakete sa više upita, upite za svime što nije ili A ili TXT ili ANY i upite za svime što nije u `BASE` domeni.

5. MaraDNS

NB: Ovaj odlomak je u nastajanju, te je moguće da sadrži greške, nepravilnosti i nepotpune informacije. Hvala na razumijevanju.

MaraDNS je još jedan poslužitelj koji pametnim dizajnom cilja imati minimalne zahtjeve za resursima i biti vrlo siguran te pouzdan softver. Za razliku od Djbdns softvera, ovaj poslužitelj se vrlo aktivno razvija i kontinuirano testira u potrazi za potencijalnim sigurnosnim i inim propustima. Također ga resi i vrlo jednostavna i čitljiva konfiguracija, pa je svega nekoliko konfiguracijskih linija dovoljno za postavljanje autoritativnog ili pak rekurzivnog poslužitelja.

Datoteke u kojima se nalaze zone imaju dva osnovna načina zapisivanja (o kojima će biti kasnije riječi), od kojih je bolje prihvaćen onaj noviji koji vrlo nalikuje na tipičnu Bind zonu. MaraDNS iznimno brzo poslužuje podatke iz svojih zona s obzirom da ih uvijek drži u memoriji, no zbog toga nije prikladno koristiti ga za izrazito velike zone. MaraDNS ne podržava NOTIFY mehanizam niti dinamičko ponovno učitavanje zone; za svaku promjenu u zoni potrebno je restartati rečeni servis. Dakle, na sekundarnim poslužiteljima je kao i kod Tinydns servisa potrebno nešto ručnog rada da bi se osposobile mogućnosti koje su npr. kod Bind poslužitelja u potpunosti automatizirane.

MaraDNS se sastoji od dva osnovna servisa:

- **Maradns** je osnovni DNS UDP poslužitelj u autoritativnom, rekurzivnom (spremičkom) ili dvojnem načinu rada. Rečeni koristi postavke iz `mararc` datoteke, učitava sve zone u memoriju i poslužuje ih. Prilikom učitavanja zone se eventualno i sintetiziraju određeni zapisi po potrebi, s obzirom da je moguće izostaviti SOA i NS zapise za pojedinu zonu. Također, moguće je i automatski generirati serijski broj u SOA, na sličan način kako to radi i Tinydns (Djbdns). Servis se radi sigurnosti uvijek chroota u radni direktorij sa zonama, otpuštajući administrativne privilegije.
- **Zoneserver** je opcionalni DNS TCP poslužitelj koji se koristi za DNS komunikaciju kroz TCP kao i prijenose zona (također TCP). Podržan je samo AXFR način prijenosa zone. Zone se poslužuju isključivo direktno s diska. Konfiguraciju također čita iz `mararc` datoteke, te se također chroota u radni direktorij sa zonama, otpuštajući administrativne privilegije.

Osim toga u uobičajenoj distribuciji dolaze i korisni popratni alati:

- **Duende** se koristi za pokretanje Maradns servisa, njegov nadzor i podizanje servisa za systemske zapisnike,
- **Askmara** je alat za jednostavne DNS upite, vrlo nalik na Dig ali dosta skromnijih mogućnosti i trivijalne sintakse,
- **Getzone** se koristi za prijenos željene zone sa udaljenog poslužitelja i prikaz u CSV1 formatu. S obzirom da osnovni servis ne podržava NOTIFY mehanizam, ovaj alat je jedan od predviđenih načina sinkronizacije sekundarnih DNS poslužitelja sa primarnim,
- **Fetchzone** se također koristi za prijenos željene zone sa udaljenog poslužitelja, no za razliku od prethodnog, prikazuje zonu u novijem

obliku zapisa, tzv. CSV2 formatu. S obzirom da je CSV2 trenutni standardni format, ovo je alat kojeg ćete vjerojatno tipično koristiti za prijenos jedne ili više zona na sekundarnom servisu. Nažalost nakon prijenosa zone osnovni DNS servis neće primijetiti promjenu u zonama niti može ponovno pročitati zone, već je nužan restart servisa.

5.1.CSV1 format zone

CSV1 tip zone je podržan od MaraDNS 1.0 inačice nadalje, no zamijenjen je nešto fleksibilnijim i čitljivijim CSV2 formatom. Ovaj tip zapisa vrlo podsjeća na Tinydns format po smanjenoj preglednosti i prilično striktnom načinu zapisivanja. U slučaju da želite postojeću DNS zonu iz nekog drugog DNS poslužitelja prebaciti u CSV1 format, najjednostavnije je učiniti to koristeći **AXFR** kroz alat `getzone`, čime odmah na standardni izlaz dobivate CSV1 zonu.

Da bi se ovaj tip zone koristio, potrebno je u `mararc` datoteku dodati:

Primjer 32: CSV1 konfiguracija za MaraDNS

```
csv1 = {}  
csv1["domena."] = "db.domena"
```

Pri tome je potrebno kao i kod drugih DNS poslužitelja paziti na završnu točku labele, dok je ime datoteke u kojoj se nalazi zona proizvoljno, ali se radi lakšeg prepoznavanja prefiksira sa "db.". Kao što je moguće vidjeti, sintaksa nalikuje na jezik Python, gdje definiramo prazni rječnik `csv1`, te zatim stavljamo u rječnik ime datoteke sa zonom, a ključ je samo ime domene.

U samoj zoni se koriste sljedeći specijalni znakovi:

- **|** (okomita crta) - međusobno odjeljuje polja u pojedinom zapisu. To je jedini dozvoljeni razdjelnik, dakle niti razmaci niti tabulatori se ne koriste, za razliku od CSV2 tipa zone.
- **#** (ljestve) - linija komentara. Rečena linija se ignorira u cijelosti, tako da je opcionalni sadržaj linije nevažan, makar je eventualno sintaksno točan.
- **%** (postotak) - u domenskim imenima označava da će biti dodano ime zone (odnosno ključ iz CSV1 rječnika). Dakle kao što se u Bind zonama koristi specijalni znak "@", tako se ovdje koristi "%".
- ***** (zvijezda) - zamjenski znak se koristi se za definiranje zamjenskih zapisa i dozvoljeno ga je pisati samo na početku pojedine labele. Preporučljivo je izbjegavati korištenje "*" zapisa sa CNAME, s obzirom da se može očekivati drukčije ponašanje od Bind9 servisa. Specifično pretpostavimo da postoji CNAME zapis za `foo.example.com`, A zapis za `*.example.com`, te ne postoji A zapis za `foo.example.com`. U slučaju da se desi A upit za `foo.example.com`, MaraDNS će vratiti A zapis za

*.example.com, dok će Bind9 vratiti NXDOMAIN. Razlog ovome je prvenstveno u različitom tumačenju RFC 1034.

- \ (obrnuta kosa crta) - omogućava da se "%" ili "\" znak nakon ne obrađuje kao poseban znak, te omogućava unošenje oktalnih vrijednosti.

Način pisanja zapisa u pojedinoj CSV1 zoni je vrlo striktan i prilično podsjeća na onaj u Tinydnsu. Greške u zoni uzrokuju da se servis odbacuje zonu s fatalnom greškom. Pravila su sljedeća:

- Svaki pojedini DNS zapis je u vlastitom redu,
- Svaki red započinje sa jednim znakom koji definira o kojem je tipu DNS zapisa riječ,
- Unutar cijelog reda postoji nekoliko odjeljaka, koji su odvojeni znakom okomite crte "|",
- Pojedini odjeljak se ne smije izostaviti.
- Razmaci i tabulatori nisu dozvoljeni, dok prazni redovi jesu,
- Svaki zapis ima vlastiti **TTL** i on ne bi trebao izostaviti. U slučaju da se to učini, bit će jednak 0 (što je sasvim različito ponašanje od onog sa CSV2 formatom zone),
- Svako pojedino domensko ime mora završavati s točkom ili sa znakom "%",
- Prije učitavanja se sva se domenska imena u zoni pretvaraju u mala slova,
- Na početku zone treba biti **SOA** zapis za domenu kojeg slijedi jedan ili više **NS** zapisa za domenu koje slijede ostali zapisi.

Tipovi zapisa su sljedeći:

- **A** (slovo A) - definira A zapise:

```
Ax.fqdn|ttl|ip
```

Rečeni zapis će stvoriti **A** zapis za labelu `x.fqdn` sa IP adresom `ip`. U slučaju da postoji više **A** zapisa oni će se u odgovoru biti po slučajnom (ne ciklički) redoslijedu.

Primjer:

```
Aesa1.esa.fer.hr.|86400|161.53.71.194  
Ahpe50.esa.fer.hr.|86400|161.53.71.235
```

- **N** (slovo N) - definira NS zapise za domenu `fqdn`. Sintaksa je sljedeća:

```
Nfqdn|ttl|x.fqdn
```

Pri tome će se u zoni pojaviti **NS** zapis `x.fqdn` kao poslužitelj za domenu `fqdn`. Ako su **NS** zapisi o autoritativnim DNS poslužiteljima za zonu, onda ih se ne smije izostaviti i oni trebaju biti na početku zone i to odmah nakon **SOA** polja.

Primjer:

```
Nesa.fer.hr.|86400|esa1.esa.fer.hr.  
Nesa.fer.hr.|86400|hpe50.esa.fer.hr.
```

- **C** (slovo C) - definira CNAME zapise:

```
Cx.fqdn|ttl|p
```

Stvara kanoničko ime odnosno **CNAME** zapis za `x.fqdn` koji pokazuje na domensko ime `p`. Nužno je izbjegavati slučajeve u kojima postoji još zapisa za `x.fqdn`. Kad god koristite **CNAME**, uvijek dobro razmislite da li vam je isti potreban ili možda možete riješiti problem sa standardnim **A** zapisom.

Primjer:

```
Cwww.esa.fer.hr.|86400|esa1.esa.fer.hr.
```

- **S** (slovo S) - definira SOA zapise za domenu `fqdn`. Sintaksa je sljedeća:

```
Sfqdn|ttl|origin|email|serial|refresh|retry|expire|  
minttl
```

Stvara u zoni **SOA** zapis za domenu `fqdn` koji ukazuje na `origin` kao primarni DNS poslužitelj i imat će `email` kao kontakt adresu, zapisanu u RFC 822 formatu (`korisnik@domena.`) s obaveznom točkom na kraju. **SOA** zapis mora biti na početku zone i smije se pojaviti samo jednom. Serijski broj `serial` se u CSV1 formatu nažalost ne može automatski generirati. Ostali parametri imaju isto značenje kao i kod ostalih DNS poslužitelja.

Primjer upotrebe bi bio sljedeći:

```
Sesa.fer.hr.|86400|esa1.esa.fer.hr.|  
postmaster@esa.fer.hr.|154140119|28800|7200|604800|  
604800
```

- **P** (slovo P) - definira PTR zapise:

```
Previp|ttl|x.fqdn
```

Stvara u zoni **PTR** zapis za IP adresu `revip` koja treba biti u reverznoj `in.addr-arpa` notaciji, a zapis pokazuje na `x.fqdn` domensko ime.

Primjer:

```
P194.71.53.161.in-addr.arpa.|86400|esa1.esa.fer.hr.  
P235.71.53.161.in-addr.arpa.|86400|hpe50.esa.fer.hr.
```

- **@** (pri) - definira MX zapise:

```
@x.fqdn|ttl|dist|mx.fqdn
```

Kroz ovaj zapis se stvara **MX** zapis `mx.fqdn` za domensko ime (ili domen) `x.fqdn` i cijenom (udaljenošću) `dist`.

Primjer upotrebe:

```
@esa.fer.hr.|86400|5|esa1.esa.fer.hr.  
@esa.fer.hr.|86400|10|hpe50.esa.fer.hr.
```

- **T** (slovo T) - definira TXT zapise:

```
Tx.fqdn|ttl|text
```

Stvara **TXT** zapis sa sadržajem `text` (može sadržavati razmake ali ne i prijelaze u idući red) za domensko ime `x.fqdn`.

Primjer:

```
Tesa.fer.hr.|86400|v=spf1 mx -all
```

- **U** (slovo U) - generički zapisi:

```
Udata|ttl|n|rdata
```

Stvara proizvoljan zapis za inače nepodržane tipove zapisa, identično kao i kod Tinydns servisa. Koristeći ovaj zapis možete definirati proizvoljan RR tipa `n` (cijeli broj između 1 i 65535) za labelu `data` sa sadržajem `rdata`. Način zapisa `rdata` se razlikuje u ovisnosti o tipu zapisa. Kao i za **TXT**, koristeći `\NNN` oktalni zapis je moguće unijeti proizvoljne znakove unutar `rdata`. Primijetite da se za `data` kod ovog specifičnog tipa zapisa ne treba dodavati točku na kraj.

Primjer:

```
#_http._tcp.esa.fer.hr. 86400 IN SRV 10 100 80  
www.esa.fer.hr  
U_http._tcp.esa.fer.hr|86400|  
33|\000\012\000\144\000\120\003www\003esa\003fer\002  
hr\000
```

5.2.CSV2 format zone

CSV2 format zone je podržan od MaraDNS 1.2 inačice nadalje. Karakteristična je veća fleksibilnost u pisanju, a sama zona svojom sintaksom prilično podsjeća na Bind zonu. Postoji i `bind2csv2.py` alat za automatiziranu konverziju Bind zona u CSV2 zone, što može poslužiti kod migracije. Druga varijanta je konverzija koristeći **AXFR** prema postojećem DNS poslužitelju. To možete učiniti koristeći alat `fetchzone`, čime na standardni izlaz dobivate CSV2 zonu.

Da bi se ovaj tip zone koristio, potrebno je u `mararc` datoteku dodati:

Primjer 33: CSV2 konfiguracija za MaraDNS

```
csv2 = {}  
csv2["domena."] = "db.domena"
```

Greške u zoni uzrokuju da se servis odbacuje zonu s fatalnom greškom. Pravila pisanja zone su nešto fleksibilnija od CSV1:

- Svaki zapis mora biti u obliku `name [+ttl] [rtype] rdata [~]`
- Sa "[...]" (uglatim zagradama) označavamo zapise koje je moguće izostaviti,
- `name` je domensko ime ili IP adresa (ili jednostavno naziv zapisa), `+ttl` je TTL koji mora započeti sa "+" (plus) znakom, `rtype` je tip zapisa (A, MX, AAAA, itd. dakle nalik na Bind sintaksu), `rdata` je sam sadržaj zapisa čiji način pisanja ovisi o tipu samog zapisa (različit oblik za npr. A, MX, SRV, itd). U slučaju da `rtype` nije zadan, podrazumijeva se **A** tip zapisa,
- Unutar svakog zapisa postoji nekoliko odjeljaka, koji su odvojeni sa "|" (okomita crta), razmacima, tabulatorima ili prelascima u idući red,
- Pojedini DNS zapis se može protezati u više redova. Kraj zapisa se može naznačiti sa znakom "~" (tilda) u MaraDNS 1.3 inačici, odnosno prelaskom u idući red,
- Svaki zapis ima vlastiti TTL i on se smije izostaviti. U slučaju da se to učini, bit će jednak 86400 rečeni zapis. To je moguće promijeniti bilo globalno bilo za grupu zapisa koristeći parametar `/ttl`, no o tome ćemo detaljnije u nastavku,
- Svako pojedino domensko ime mora završavati s točkom ili sa znakom "%",
- Vrijede isti specijalni znakovi kao i u CSV1 formatu zone,
- Vrijede ista pravila za komentare kao i u CSV1 formatu, osim što u komentarima nije dozvoljen znak "{" (otvorena vitičasta zagrada) te znak "'" (jednostruki navodnik),
- U pojedinim poljima i kod pojedinih tipova zapisa moguće je koristiti dodatne parametre odnosno naredbe o kojima ćemo detaljnije u nastavku,
- Znak "~" (tilda) se ne bi trebao pojavljivati u zoni ako se ne koristi za odvajanje zapisa,
- Prije učitavanja se sva se domenska imena u zoni pretvaraju u mala slova,
- Na početku zone treba biti **SOA** zapis za domenu kojeg slijedi jedan ili više **NS** zapisa za domenu koje slijede ostali zapisi. U slučaju da rečeni zapisi ne postoje, biti će generirani automatski tako da zona može normalno funkcionirati.

Tipovi zapisa su ukratko sljedeći (za više informacija, pogledajte Bind9 poglavlje):

- **A** - definira A zapise:

```
x.fqdn +ttl A ip
```

Rečeni zapis će stvoriti **A** zapis za labelu `x.fqdn` sa IP adresom `ip`. U slučaju da postoji više **A** zapisa oni će se u odgovoru biti po slučajnom (ne ciklički) redoslijedu. Ovo je podrazumijevani tip zapisa, pa je njegovu oznaku moguće izostaviti.

Primjer:

```
esa1.esa.fer.hr. a 161.53.71.194
hpe50.% 161.53.71.235
```

- **PTR** - definira PTR zapise:

```
revip +ttl PTR x.fqdn
```

Stvara u zoni **PTR** zapis za IP adresu `revip` koja treba biti u reverznoj `in.addr-arpa` notaciji, a zapis pokazuje na `x.fqdn` domensko ime.

Primjer:

```
194.71.53.161.in-addr.arpa. ptr esa1.esa.fer.hr.
235.71.53.161.in-addr.arpa. ptr hpe50.esa.fer.hr.
```

- **MX** - definira MX zapise:

```
x.fqdn +ttl MX dist mx.fqdn
```

Kroz ovaj zapis se stvara **MX** zapis `mx.fqdn` za domensko ime (ili domenu) `x.fqdn` i cijenom (udaljenošću) `dist`.

Primjer upotrebe:

```
esa.fer.hr. mx 5 esa1.esa.fer.hr.
esa.fer.hr. mx 10 hpe50.esa.fer.hr.
```

- **AAAA** - definira AAAA zapise:

```
x.fqdn +ttl AAAA ip6
```

Rečeni zapis će stvoriti **AAAA** zapis za labelu `x.fqdn` sa IPv6 adresom `ip6`, koja treba biti u tipičnom IPv6 obliku prema RFC 4291 (8 16-bitnih brojeva odvojenih dvotočkama).

Primjer:

```
ipv6.carnet.hr. aaaa 2001:b68:e160:0:0:0:0:25
```

- **SRV** - definira SRV zapise:

```
_serv._proto.fqdn +ttl SRV prio weight x.fqdn
```


Rečeni zapis će stvoriti **SRV** zapis za određeni servis `_serv` (lista mogućih servisa je u RFC 2782), tip protokola `_proto` i za labelu `fqdn`. Polje `prio` je prioritet, `weight` je odgovarajuća težina te `x.fqdn` je cilj odnosno meta na koju sam zapis pokazuje.

Primjer:

```
_sip._tcp.srce.hr. srv 0 0 5060 ser.srce.hr.  
_sip._udp.srce.hr. srv 0 0 5060 ser.srce.hr.
```

- **NS** - definira NS zapise za domenu `fqdn`. Sintaksa je sljedeća:

```
fqdn +ttl NS x.fqdn
```

Pri tome će se u zoni pojaviti **NS** zapis `x.fqdn` kao poslužitelj za domenu `fqdn`. Ako su **NS** zapisi o autoritativnim DNS poslužiteljima za zonu, oni trebaju biti na početku zone ili odmah nakon **SOA** polja. Ako se izostave, biti će automatski generirani koristeći javne adrese na kojima sam servis osluškuje upite.

Primjer:

```
esa.fer.hr. ns esa1.esa.fer.hr.  
esa.fer.hr. ns hpe50.esa.fer.hr.
```

- **SOA** - definira SOA zapise za domenu `fqdn`. Sintaksa je sljedeća:

```
fqdn +ttl SOA origin email serial refresh retry  
expire minttl
```

Stvara u zoni **SOA** zapis za domenu `fqdn` koji ukazuje na `origin` kao primarni DNS poslužitelj i imat će `email` kao kontakt adresu, zapisanu u RFC 822 formatu (`korisnik@domena.`) s obaveznom točkom na kraju. **SOA** zapis mora biti na početku zone i smije se pojaviti samo jednom. U slučaju da se izostavi, biti će automatski generiran. Umjesto serijskog broja se može koristiti `/serial` parametar koji omogućava generiranje serijskog broja koristeći vrijeme stvaranja datoteke u kojoj se nalazi zona (slično kao i kod Tinydns servisa), čime se garantira da serijski broj prati promjene u zoni. Ostali parametri imaju isto značenje kao i kod ostalih DNS poslužitelja.

Primjer upotrebe bi bio sljedeći:

```
esa.fer.hr. soa esa1.esa.fer.hr.  
postmaster@esa.fer.hr. /serial 28800 7200 604800  
604800
```

- **TXT** - definira TXT zapise:

```
x.fqdn +ttl TXT text
```

Stvara **TXT** zapis sa sadržajem `text` (može sadržavati razmake ali ne i prijelaze u idući red) za domensko ime `x.fqdn`. Ako `text` sadrži samo ASCII ili UTF-8 kodirane znakove, može se jednostavno navesti unutar jednostrukih navodnika, kao u primjeru. Druga varijanta je koristiti znak `\xNN` odnosno `\ONNN` za heksadecimalnu (dvoznamenkasti broj) odnosno oktalnu (troznamenkasti broj) specifikaciju pojedinog znaka, te pri tome nije potrebno tekst okružiti jednostrukim navodnicima.

Primjer:

```
igh.hr. txt 'v=spf1 mx -all'
g.example.com. txt \200\x81\202\x83
```

- **SPF** - definira SPF zapise. Rečeni su identični **TXT** polju, a sadrže informacije iz RFC 4408.
- **RAW** - generički zapisi:

```
data +ttl RAW n rdata
```

Stvara proizvoljan zapis za inače nepodržane tipove zapisa, identično kao i kod Tinydns servisa. Koristeći ovaj zapis možete definirati proizvoljan RR tipa `n` (cijeli broj između 1 i 65535) za labelu `data` sa sadržajem `rdata`. Način zapisa `rdata` se razlikuje u ovisnosti o tipu zapisa. Kao i za **TXT**, ako `text` sadrži samo ASCII ili UTF-8 kodirane znakove, može se jednostavno navesti unutar jednostrukih navodnika, kao u primjeru. Druga varijanta je koristiti znak `\xNN` odnosno `\ONNN` za heksadecimalnu (dvoznamenkasti broj) odnosno oktalnu (troznamenkasti broj) specifikaciju pojedinog znaka, te pri tome nije potrebno tekst okružiti jednostrukim navodnicima.

Primjer:

```
_sip._tcp.srce.hr. raw 33
\x00\x00\x00\x00\x13\xc4\x03'ser'\x04'srce'\x02'hr'\
x00
ipv6.carnet.hr. raw 28 '
'\x01\x0b'h'\xe1'``'\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00%'
```

- **FQDN4** - definira A+PTR zapise:

```
x.fqdn +ttl FQDN4 ip
```

Rečeni zapis će stvoriti **A** zapis za labelu `x.fqdn` sa IP adresom `ip` te istovremeno i odgovarajući **PTR** zapis koji pokazuje prema `x.fqdn`. Kod ovakvog tipa zapisa se uglavnom treba izbjegavati višestruko korištenje za istu labelu, s obzirom da će to uzrokovati uz višestruke **A** zapise i višestruke **PTR** zapise za rečenu labelu.

Primjer:

```
esal.esa.fer.hr. fqdn4 161.53.71.194
hpe50.esa.fer.hr. fqdn4 161.53.71.235
```

- **FQDN6** - definira AAAA+PTR zapise:

```
x.fqdn +ttl FQDN6 ip6
```

Rečeni zapis će stvoriti **AAAA** zapis za labelu `x.fqdn` sa IPv6 adresom `ip6`, koja treba biti u tipičnom IPv6 obliku prema RFC 4291 (8 16-bitnih brojeva odvojenih dvotočkama). Kod ovakvog tipa zapisa se uglavnom treba izbjegavati višestruko korištenje za istu labelu, s obzirom da će to uzrokovati uz višestruke **AAAA** zapise i višestruke **PTR** zapise za rečenu labelu.

Primjer:

```
ipv6.carnet.hr. fqdn6 2001:b68:e160:0:0:0:0:25
```

CNAME - definira CNAME zapise:

```
x.fqdn +ttl CNAME p
```

Stvara kanoničko ime odnosno **CNAME** zapis za `x.fqdn` koji pokazuje na domensko ime `p`. Nužno je izbjegavati slučajeve u kojima postoji još zapisa za `x.fqdn`. Kad god koristite **CNAME**, uvijek dobro razmislite da li vam je isti potreban ili možda možete riješiti problem sa standardnim **A** zapisom.

Primjer:

```
www.esa.fer.hr. cname esal.esa.fer.hr.
```

Uz navedene zapise i specijalne znakove, ponegdje je moguće koristiti i specijalne naredbe odnosno parametre sa posebnim značenjem:

- `/serial` - u **SOA** polju uzrokuje automatsko generiranje serijskog broja za zonu.

Primjer:

```
esa.fer.hr. soa esal.esa.fer.hr.
postmaster@esa.fer.hr. /serial 28800 7200 604800
604800
```

- `/ttl` - uzrokuje da se za zapise ispod rečenog parametra mijenja standardna TTL vrijednost. Kao što smo rekli, ako se TTL izostavi on ima tipično vrijednost 86400, međutim s rečenom naredbom je moguće grupno postaviti TTL zapisima na željenu vrijednost.

Primjer:

```
a.x.com. a 10.0.0.1 # a.x.com. +86400 a 10.0.0.1
/ttl 3600
b.x.com. a 10.0.0.2 # b.x.com. +3600 a 10.0.0.2
```

```
c.x.com. +9600 a 10.0.0.3 # ...  
/ttl 7200  
d.x.com. a 10.0.0.4 # d.x.com. +7200 a 10.0.0.4
```

- `/origin` - služi mijenjanju sufiksa (tipično to je ime zone) koji se koristi za zamjenu znaka "%" u zoni. Dakle, rečena naredba neće promijeniti stvarno ime domene koje određuje jesu li zapisi u domeni autoritativni, već služi samo za mijenjanje vrijednosti za rečeni zamjenski znak.

Primjer:

```
/origin x.com. # % = x.com.  
www.% 10.1.0.1 # www.x.com. a 10.1.0.1  
% mx 10 mail.% # x.com. mx 10 mail.x.com.  
mail.% 10.1.0.2 # mail.x.com. a 10.1.0.2  
/origin x.org. # % = x.org.  
www.% 10.2.0.1 # www.x.org. a 10.2.0.1  
% mx 10 mail.% # x.org. mx 10 mail.x.org.  
mail.% 10.2.0.2 # mail.x.org. a 10.2.0.2
```

- `/opush` - omogućava spremanje zadnje vrijednosti za znak "%" na stog. Moguće je spremiti do 7 vrijednosti.
- `/opop` - omogućava dohvat zadnje vrijednosti za znak "%" sa stoga, skida rečenu vrijednost sa stoga i postavlja ju kao novi sufiks za znak "%", kao što radi `/origin` naredba.

Primjer:

```
/origin x.com. # % = x.com.; prazan stog  
/opush mail.% # % = mail.x.com; x.com na stogu  
a.% 10.4.0.1 # a.mail.x.com a 10.4.0.1  
/opush web.x.com. # mail.x.com i x.com na stogu  
a.% 10.5.0.1 # a.web.x.com a 10.5.0.1  
b.% 10.5.0.2 # b.web.x.com a 10.5.0.2  
/opop # % = mail.x.com; x.com na stogu  
b.% 10.4.0.2 # b.mail.x.com a 10.4.0.2  
/opop # % = x.com.; prazan stog  
% mx 10 a.mail.% # x.com. mx 10 a.mail.x.com.  
% mx 20 b.mail.% # x.com. mx 20 b.mail.x.com.
```

- `/read` - uzrokuje da se učitava sadržaj navedene datoteke kao da je dio zone.

Primjer:

```
/origin foo.x.com.  
% txt 'Foomatic!'  
/read datoteka  
% mx 10 mail.foo.x.com.
```

U slučaju da rečena datoteka mijenja vrijednost za "%", potrebno je

koristiti sljedeće:

```
/opush %  
/read datoteka  
/opop
```

5.3.Mararc konfiguracijska datoteka

Poglavlje u nastanku...

6. PowerDNS

NB: Ovaj odlomak je u nastajanju, te je moguće da sadrži greške, nepravilnosti i nepotpune informacije. Hvala na razumijevanju.

7. Unbound

NB: Ovaj odlomak je u nastajanju, te je moguće da sadrži greške, nepravilnosti i nepotpune informacije. Hvala na razumijevanju.

8. Dnsmasq

NB: Ovaj odlomak je u nastajanju, te je moguće da sadrži greške, nepravilnosti i nepotpune informacije. Hvala na razumijevanju.

9. NSD

NB: Ovaj odlomak je u nastajanju, te je moguće da sadrži greške, nepravilnosti i nepotpune informacije. Hvala na razumijevanju.

10.Primjeri konfiguracija

Donosimo različite dijelove konfiguracije stvarnih DNS poslužitelja. Prije ikakve upotrebe preporuča se pročitati prethodna poglavlja. Naravno, ovo su tek dijelovi konfiguracije koji mogu ali i ne moraju u funkcionirati - već trebaju poslužiti samo kao primjer za pisanje vlastitih konfiguracija.

10.1.Bind9 konfiguracija - *named.conf*

```
// kome dajemo zone transfer
// primijetite -- poželjno je davati zone transfer
// samo nadređenim DNS poslužiteljima
acl "xfer" {
    161.53.72.21;
    161.53.3.7;
    161.53.2.69;
    161.53.2.70;
    161.53.123.3;
    161.53.116.9;
    161.53.71.194;
    127.0.0.1;
    161.53.97.3;
    161.53.97.11;
};

// kome dajemo rekurziju
// primijetite -- poželjno je davati uslugu
// rekurzije samo računalima iz vlastite mreže!
acl "trusted" {
    161.53.116.0/22;
    193.198.206.0/24;
    193.198.217.192/27;
    localhost;
};

// parametri rada
options {
    directory "/etc/bind";
    query-source address * port 53;
    allow-transfer { xfer; };
    allow-recursion { trusted; };
    version "Unknown";
    transfer-format many-answers;
    max-transfer-time-in 120;
    interface-interval 120;
    notify yes;
    recursion yes;
    minimal-responses yes;
    notify-source 161.53.116.8;
    transfer-source 161.53.116.8;
};
```

```
// ugasi lame-servers u logovima
logging {
    category lame-servers { null; };
};

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

// localhost domena
zone "localhost" {
    type master;
    file "/etc/bind/db.local";
};

// reverse 127
zone "127.in-addr.arpa" {
    type master;
    file "/etc/bind/db.127";
};

// reverse 0
zone "0.in-addr.arpa" {
    type master;
    file "/etc/bind/db.0";
};

// reverse 255
zone "255.in-addr.arpa" {
    type master;
    file "/etc/bind/db.255";
};

// nas forward
zone "fsb.hr" {
    type master;
    file "/etc/bind/hosts_fsb.db";
};

// nasi reverseovi
zone "116.53.161.in-addr.arpa" {
    type master;
    file "/etc/bind/hosts_116.rev";
};

// itd.

// secondary forward
```

```
zone "fpz.hr" {
    type slave;
    file "/etc/bind/hosts_fpz.db";
    masters { 161.53.97.3; 161.53.97.11; };
};

// itd.

// razne autogenerirane stvari
include "/etc/bind/zones.rfc1918";
include "/etc/bind/spywaredomains.zones";
```

10.2.Bind9 forward zona - hosts_fsb.db

```
; normalna forward zona
$TTL 1D
@ SOA      localhost.fsb.hr. postmaster.fsb.hr. (
                                200508241      ; Serial
                                28800            ; Refresh - 5 minutes
                                7200            ; Retry - 1 minute
                                604800          ; Expire - 2 weeks
                                86400 )         ; Minimum - 12 hours
    NS     hobbit.fsb.hr.
    NS     bjesomar.srce.hr.
    NS     mafpz.fpz.hr.
    MX     5      hobbit
    A      161.53.116.9
    TXT    "v=spf1 ip4:161.53.116.0/22
ip4:193.198.206.0/24 ip4:193.198.217.192/27 a mx ptr
~all"

localhost      A      127.0.0.1
               AAAA   ::1
cisco           A      161.53.116.1

sw3404-rc-1          A      161.53.116.2
sw404-rc-2           A      161.53.116.3
sw404-rc-3           A      161.53.116.4
; itd.
arwen              A      161.53.116.15
fsbwireless.fsb.hr. CNAME   arwen.fsb.hr.
wireless.fsb.hr.   CNAME   arwen.fsb.hr.
; itd.
```

10.3.Bind9 reverse zona - db.127

```
; reverzna zona za loopback
$TTL      604800
@ IN      SOA      localhost. root.localhost. (
                                1      ; Serial
```

```

                                604800      ; Refresh
                                86400       ; Retry
                                2419200    ; Expire
                                604800 )   ; Negative Cache
TTL
;
@      IN      NS      localhost.
1.0.0  IN      PTR     localhost.
```

10.4.Bind9 wildcard zona - blockeddomain.hosts

```

; sve moguće zapise u zoni preusmjerava na 127.0.0.1
; efikasno za blokiranje cijelih domena za pojedinu
; ustanovu
$TTL      86400      ; one day
@          SOA      ns0.bleedingsnort.com.
bleeding.bleedingsnort.com. (
                                1
                                28800     ; refresh  8 hours
                                7200      ; retry    2 hours
                                864000    ; expire   10 days
                                86400 )   ; min ttl  1 day
                                NS        ns0.bleedingsnort.com.
                                NS        ns1.bleedingsnort.com.
                                A          127.0.0.1
*          A          127.0.0.1
```

10.5.Bind9 prazna zona - db.empty

```

; ovdje nema ničega
$TTL      86400
@          IN      SOA      localhost. root.localhost. (
                                1          ; Serial
                                604800     ; Refresh
                                86400      ; Retry
                                2419200    ; Expire
                                86400 )   ; Negative Cache
TTL
;
@          IN      NS      localhost.
```

10.6.Bind9 reverse zona - hosts_116.rev

```

; normalna reverzna zona
TTL 1D
@          SOA      localhost.fsb.hr.
postmaster.fsb.hr. (
                                200508234 ; Serial
                                28800      ; Refresh - 5
minutes
```

```
minute          7200          ; Retry - 1
weeks           604800        ; Expire - 2
hours           86400 )       ; Minimum - 12
NS              hobbit.fsb.hr.
NS              bjesomar.srce.hr.
NS              mafpz.fpz.hr.
1 PTR           cisco.fsb.hr.
2 PTR           sw3404-rc-1.fsb.hr.
3 PTR           sw404-rc-2.fsb.hr.
4 PTR           sw404-rc-3.fsb.hr.
5 PTR           fsb-backrout.fsb.hr.
; itd.
```

10.7.Bind9 MS Active Directory kompatibilna zona

```
; MS domena je terminator.local
; njoj treba odgovarati zona u Bind9 konfiguraciji
; neka je DNS poslužitelj ns1 sa IP adresom 192.168.16.1
; neka je MS AD poslužitelj terminator-1234 sa IP adresom
; 192.168.16.3
$TTL 86400
@      SOA      ns1 hostmaster.terminator-1234 (
        1      ; Serial
        28800   ; Refresh
        7200    ; Retry
        604800  ; Expire
        86400 ) ; Minimum
        NS      ns1
        A       192.168.16.3
localhost      A       127.0.0.1
ns1             A       192.168.16.1
terminator      A       192.168.16.3
terminator-1234 A       192.168.16.3
_ldap._tcp      SRV     0 0 389 terminator-1234
_ldap._tcp.dc._msdcs SRV 0 0 389 terminator-1234
_ldap._tcp.pdc._msdcs SRV 0 0 389 terminator-1234
_kerberos._tcp  SRV     0 0 88  terminator-1234
_kerberos._udp  SRV     0 0 88  terminator-1234
_kerberos._tcp.dc._msdcs SRV 0 0 88  terminator-1234
_kerberos._udp.dc._msdcs SRV 0 0 88  terminator-1234
gc._msdcs       SRV     0 0 3268 terminator-1234
; odgovarajuća konfiguracija Bind9 named.conf
; zone "terminator.local" {
;     type master;
;     file "/etc/bind/hosts_terminator.db";
;     check-names ignore;
;     allow-update { 192.168.16.3; };
; }
```

10.8. TinyDNS zona

```
&hybserv.net::dns.hybserv.net.:86400
&hybserv.net::ns.icsbg.net.:86400
+cvns.hybserv.net:161.53.71.235:86400
+dns.hybserv.net:161.53.71.235:86400
+hybserv.net:161.53.71.235:86400
+localhost.hybserv.net:127.0.0.1:86400
+mail.hybserv.net:161.53.71.235:86400
+www.hybserv.net:161.53.71.235:86400
@hybserv.net::dns.hybserv.net.:5:86400
@hybserv.net::ns.icsbg.net.:10:86400
Csvn.hybserv.net:cvns.hybserv.net.:86400
Ctrac.hybserv.net:cvns.hybserv.net.:86400
Cviewcvns.hybserv.net:cvns.hybserv.net.:86400
Cw.hybserv.net:www.hybserv.net.:86400
Cweb.hybserv.net:www.hybserv.net.:86400
Cww.hybserv.net:www.hybserv.net.:86400
Zhybserv.net:dns.hybserv.net.:postmaster.hybserv.net.:28
800:7200:604800:604800:86400
```

10.9. MaraDNS CSV1 zona

```
Shybserv.net.|86400|dns.hybserv.net.|
postmaster@hybserv.net.|154140119|28800|7200|
604800|604800
Nhybserv.net.|86400|dns.hybserv.net.
Nhybserv.net.|86400|ns.icsbg.net.
@hybserv.net.|86400|5|dns.hybserv.net.
@hybserv.net.|86400|10|ns.icsbg.net.
Acvns.hybserv.net.|86400|161.53.71.235
Adns.hybserv.net.|86400|161.53.71.235
Ahybserv.net.|86400|161.53.71.235
Awww.hybserv.net.|86400|161.53.71.235
Csvn.hybserv.net.|86400|cvns.hybserv.net.
Ctrac.hybserv.net.|86400|cvns.hybserv.net.
Cw.hybserv.net.|86400|www.hybserv.net.
Cweb.hybserv.net.|86400|www.hybserv.net.
Cww.hybserv.net.|86400|www.hybserv.net.
```

10.10. MaraDNS CSV2 zona

```
hybserv.net. soa dns.hybserv.net. postmaster@hybserv.net.
/serial 28800 7200 604800 604800
hybserv.net. ns dns.hybserv.net.
hybserv.net. ns ns.icsbg.net.
hybserv.net. mx 5 dns.hybserv.net.
hybserv.net. mx 10 ns.icsbg.net.
cvns.hybserv.net. a 161.53.71.235
dns.hybserv.net. a 161.53.71.235
hybserv.net. a 161.53.71.235
```

```
www.hybserv.net. a 161.53.71.235
svn.hybserv.net. cname cvs.hybserv.net.
trac.hybserv.net. cname cvs.hybserv.net.
w.hybserv.net. cname www.hybserv.net.
web.hybserv.net. cname www.hybserv.net.
ww.hybserv.net. cname www.hybserv.net.
```


11.Literatura

- ISO 3166, ISO 3166-1 Alpha-2, ISO 3166-3
- RFC 822: Domain names: Concepts and facilities
- RFC 823: Domain names: Implementation and specification
- RFC 974: Mail routing and the domain system
- RFC 1034: Domain names: Concepts and facilities
- RFC 1035: Domain names: Implementation and specification
- RFC 1101: DNS encoding of the network names and other types
- RFC 1123: Requirements for Internet Hosts - application and support
- RFC 1123: Requirements for Internet hosts: application and support
- RFC 1183: New DNS RR definitions
- RFC 1394: Relationship between Internet domain names and telex ID codes
- RFC 1464: Using the Domain Name System to store arbitrary string attributes
- RFC 1535: A security problem and proposed correction with widely deployed DNS software
- RFC 1536: Common DNS implementation errors and suggested fixes
- RFC 1537: Common DNS data file configuration errors
- RFC 1591: Domain name system structure and delegation
- RFC 1706: DNS NSAP resource records
- RFC 1713: Tools for DNS debugging
- RFC 1794: DNS support for load balancing
- RFC 1876: A means for expressing location information in the Domain Name System
- RFC 1886: DNS extensions to support IP version 6
- RFC 1912: Common DNS operation and configuration errors
- RFC 1918: Address Allocation for Private Internets
- RFC 1982: Serial number arithmetic
- RFC 1995: Incremental zone transfers in DNS
- RFC 1996: A mechanism for prompt notification of zone changes
- RFC 2010: Operational criteria for root name servers
- RFC 2052: A DNS RR for specifying the location of services
- RFC 2065: Domain name system security extensions
- RFC 2136: Dynamic updates in the domain name system
- RFC 2137: Secure domain name system dynamic update
- RFC 2163: Using the Internet DNS to distribute MIXER conformant global address mapping
- RFC 2168: Resolution of Uniform Resource Identifiers using the Domain Name System
- RFC 2181: Clarifications to the DNS specification
- RFC 2219: Use of DNS aliases for network services
- RFC 2230: Key exchange delegation record for the DNS
- RFC 2240: A legal basis for domain name allocation
- RFC 2308: Negative caching of DNS queries
- RFC 2317: Classless IN-ADDR.ARPA delegation

- RFC 2345: Domain names and company name retrieval
- RFC 2352: A convention for using legal names as domain names
- RFC 2671: Extension Mechanisms for DNS (EDNS0)
- RFC 2782: A DNS RR for specifying the location of services (DNS SRV)
- RFC 2845: Secret key transaction authentication for DNS (TSIG)
- RFC 2870: Root Name Server Operational Requirements
- RFC 3330: Special-Use IPv4 Addresses
- RFC 3425: Obsoleting IQUERY
- RFC 3596: DNS Extensions to Support IP Version 6
- RFC 3912: Whois protocol specification
- RFC 4697: Observed DNS Resolution Misbehavior
- Wikipedia - <http://www.wikipedia.org/>
- The TCP/IP Guide - <http://www.tcpipguide.com/>
- Cisco: Configuring the DNS Service
- Cricket Liu, Paul Albitz: DNS and BIND in a Nutshell
- DNS report - <http://www.dnsreport.com/>
- Life with djbdns - <http://www.lifewithdjbdns.com/>
- Tinydns.Org - <http://www.tinydns.org/>
- Root-servers.Org - <http://www.root-servers.org/>
- Duane Wessels: [Is Your Caching Resolver Polluting the Internet?](#)
- Duane Wessels: [Wow, That's a Lot of Packets](#)
- Duane Wessels: [Measurements and Laboratory Simulations of the Upper DNS Hierarchy](#)
- Steve Gibbard: [Geographic Implications of DNS Infrastructure Distribution](#)
- ISC OARC - <http://oarc.isc.org/>