UNIVERZITET U BEOGRADU
ELEKTROTEHNIČKI FAKULTET

Studija operativnog rada DNS servisa

Preporuke za optimalno konfigurisanje sa osvrtom na Internet bezbednost

Autor: mr Nenad Krajnović, dipl. inž.

Beograd, 2015.

1 Sadržaj

2 Uvod

Komunikacija na Internetu zasniva se na poznavanju IP adrese druge strane sa kojom je potrebno komunicirati. Pošto je ljudima jednostavnije da pamte nazive u tekstualnoj formi nego gomilu brojeva, uveden je DNS (*Domain Name Service*) Internet servis čiji je primarni zadatak da na osnovu imena odredi IP adresu, i obrnutno, na osnovu IP adrese da odredi ime hosta na Internetu. Kao i većina ostalih Internet servisa, zasniva se na klijent-server principu. Klijentski deo DNS-a čini *resolver*, koji je sastavni deo mrežnog softvera svih uređaja povezanih na Internet. Zadatak *resolver*-a je slanje upita tipa "Koja je IP adresa računara www.rnids.rs?". Pored ovog osnovnog pitanja, *resolver* može da zahteva od DNS servera i veliki broj drugih podataka koji se nalaze u DNS bazi. Serverski deo servisa nalazi se na DNS serverima i ima zadatak da na osnovu upita pripremi i pošalje odgovor sa traženim podacima. Prilikom korišćenja svakog Internet servisa, prvi korak je da se pomoću DNS servisa odredi IP adresa druge strane u komunikaciji. Zbog toga je ispravno funkcionisanje DNS servisa od ključne važnosti za rad celokupnog Interneta.

S druge strane, pitanjem sigurnosti i stabilnosti funkcionisanja Interneta u velikoj se meri bave i CERT (*Computer Emergence Response Team*) organizacije u svetu. Ako se pogleda lista zadataka CERT-a, vidi se da obuka i publikovanje preporuka za sigurno funkcionisanje Internet servisa predstavljaju dve ključne aktivnosti. U tom smislu, i ova preporuka za sigurno funkcionisanje DNS servisa se može posmatrati i kao određeni vid CERT preporuke.

3 Osnove DNS servisa

Prema istraživanju koje je sproveo ISC (*Internet Software Consortium*), broj hostova na Internetu, koje prepoznaje DNS servis, je premašio jednu milijardu [1]. Čuvanje tolike količine podataka na jednom mestu bi predstavljalo ozbiljan tehnički izazov. Pri tome, treba imati u vidu da u svakom trenutku postoji ogroman broj zahteva za dobijanjem nekog od navedenih podataka. S druge strane, navedeni podaci nisu statični, već se konstantno menjaju u manjoj ili većoj meri. Polazeći od potrebnih funkcionalnih karakteristika DNS-a, DNS servis je zamišljen i implementiran kao globalno distribuirana, skalabilna, hijerarhijski organizovana, dinamička baza podataka. Navedeni način organizacije DNS-a omogućava da se tako velika količina podataka distribuira na veliki broj servera. Svaki od servera kod sebe čuva samo delić infomacija koje čine DNS bazu podataka. Pored toga što lokalni DNS serveri čuvaju podatke, oni su odgovorni i za njihovu ažurnost. To znači da se podaci lokalno formiraju i ažuriraju, ali su globalno dostupni.

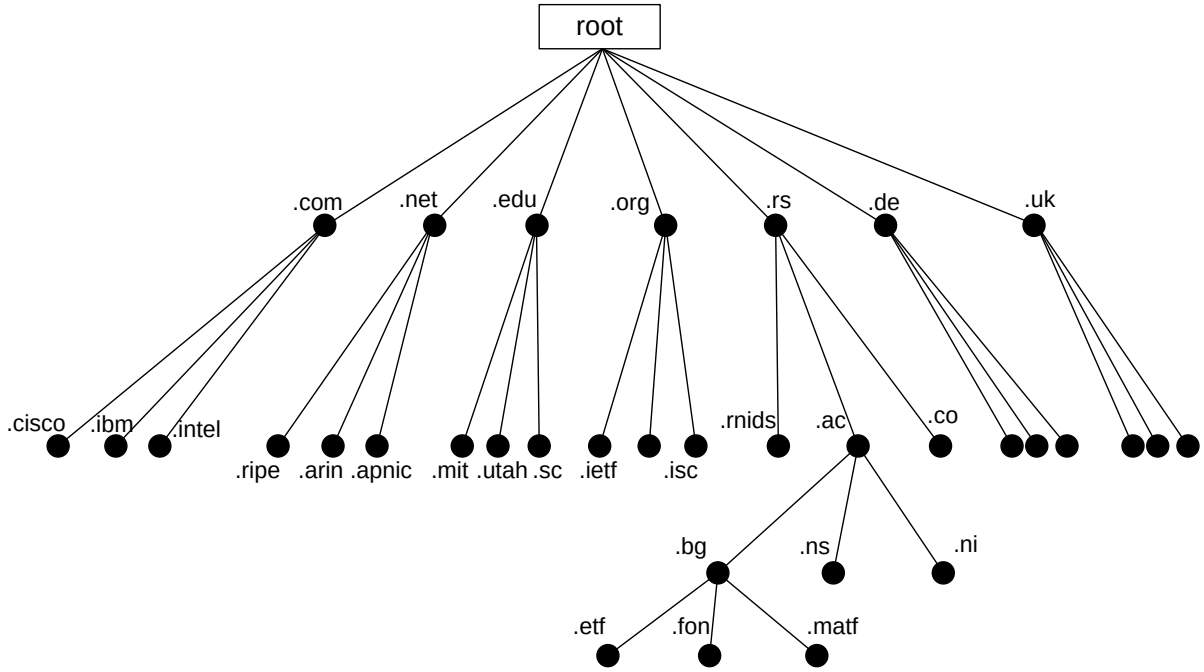
Imajući u vidu konstantno širenje Interneta, DNS baza je u startu realizovana kao skalabilna, bez ograničenja po pitanju veličine.

DNS je hijerarhijski organizovan da bi mogao da obezbedi distribuiranost i skalabilnost. Hijerarhijska organizacija DNS-a je u skladu sa hijerarhijskom organizacijom domenskog prostora. Jedan od razloga za uvođenje DNS servisa je bio i nedostatak različitih imena za nove hostove koji su se povezivali na Internet. Da bi se prevazišao taj problem, autor DNS-a, Paul Mockapetris, je definisao domenski prostor kao stablo koje počinje od *root* naziva, koji je obeležen tačkom ".", i gde svaka grana ima svoje ime - domen, što je predstavljeno na slici 1. Naziv domena se sastoji iz jedne ili više labela koje su odvojene tačkom ".". Prema RFC-u 952, labele se sastoje iz slova engleskog alfabeta i brojeva - [A...Z,0...9] i ne pravi se razlika između velikih i malih slova.

Dodela domena se odvija na hijerarhijski način, ekvivalentno sa samom strukturom domenskog prostora koja je predstavljena na slici 1. ICANN (*Internet Corporation for Assigned Names and Numbers*) je organizacija koja je nadležna za *root* domen. Ova organizacija, u skladu sa hijerarhijom domenskog prostora i usvojenim pravilama, kreira poddomene *root* domena, koje nazivamo *Top Level* Domenima ili skraćeno TLD. Prilikom kreiranja TLD-a, ICANN ujedno definiše i organizaciju koja će biti zadužena za ovaj domen. Zadatak organizacije koja je nadležna za neki domen je:

- održava minimalno dva DNS servera za domen,
- brine o tome da svi podaci koji se nalaze u DNS serveru budu ažurni,
- po potrebi definiše politiku otvaranja poddomena navedenog domena.

TLD-ove se dele na generičke (poput .com, .info,...) i *Country Code* TLD-ove (CCTLD) poput domena Srbije (.rs). Svaki vlasnik domena može da kreira neograničen broj poddomena. Prilikom kreiranja poddomena, vlasnik domena može da delegira upravljaje poddomenom nekoj drugoj organizaciji ili da on sam nastavi administraciju tog poddomena.



Slika 1 - Izgled dela domenskog prostora Interneta

Ako se pogleda hijerarhijska organizacija domenskog prostora, primeri korektnih naziva domena su rnids.rs, etf.bg.ac.rs, ietf.org, ripe.net... Maksimalna dužina domena ne sme da prelazi 225 karaktera, dok dužina jedne labele (polje između dve tačke u nazivu domena) ne sme da bude duže od 63 karaktera. Osim toga, ne postoji drugo ograničenje po pitanju dubine kreiranja domenskog stabla.

4 Način rada DNS servisa

4.1 Autoritativni i neautoritativni DNS serveri

Kada govorimo o DNS serverima, možemo ih svrstati u dve grupe, autoritativne i neautoritativne DNS servere. Autoritativni DNS serveri su oni serveri koji podatke o domenima preuzimaju iz lokalnih fajlova. Pošto ne postoji mogućnost kompromitovanja tih podataka, smatra se da su oni izvor tačnih podataka o navedenom domenu i zbog toga se nazivaju autoritativni DNS serveri.

Neautoritativni serveri su svi ostali serveri koji do podataka o domenima/hostovima dolaze na osnovu informacija koje dobijaju od autoritativnih DNS servera. U opštem slučaju, neautoritativni DNS serveri mogu i međusobno da razmenjuju informacije o domenima/hostovima.

Za svaki domen koji je predstavljen na slici 1, postoje dva ili više autoritativnih DNS servera. Njihov zadatak je da čuvaju kod sebe sve podatke koji se odnose na navedeni domen kao i da odgovaraju na upite koji se odnose na navedene podatke. Sami podaci, koji su organizovani u formi *Resource Record*-a (RR), čuvaju se u tzv. zonskom fajlu. Svaki domen ima svoj odgovarajući zonski fajl (postoji mogućnost smeštanja podataka za više domena u jedan isti fajl, ali takvo rešenje nije praktično i lako može da dovede do zabune).

Autoritativne DNS servere delimo na primarni (jedan) i sekundarne (jedan ili više) DNS servere. Podela na primarni i sekundarne DNS servere je relevantna za proces ažuriranja podataka u tabelama DNS servera. Podaci se direktno ažuriraju samo na primarnom DNS serveru, posle čega primarni DNS automatski ažurira podatke u tabelama na sekundardnim DNS serverima. Proces ažuriranja podataka u tabelama DNS servera se obavlja tako što se novi podaci upisuju u tabelu na primarnom DNS serveru. Po učitavanju novih podataka za određeni domen, primarni DNS server šalje *NOTIFY* poruku (RFC 1996) svim sekundarnim DNS serverima. Po dobijanju *NOTIFY* poruke, sekundarni DNS serveri će uraditi jednu od dve moguće stvari: zahtevaće prenos kompletne nove tabele za navedeni domen (AXFR) ili će zahtevati prenos samo novih podataka u tabeli za navedeni domen (IXFR - RFC 1995). Na taj način se obezbeđuje sinhronizacija podataka u primarnom i sekundardnim DNS serverima. Pored prethodno navedenog načina za iniciranje transfera novog zonskog fajla na sekundarni DNS server, postoji još jedan način. U okviru SOA RR-a, postoji tajmer koji se zove *refresh* tajmer i koji definiše posle kog vremena će sekundarni DNS server da proveri na primarnom DNS serveru da li je došlo do povećanja serijskog broja u SOA RR-u. Provera se obavlja tako što sekundarni DNS server pošalje upit primarnom DNS serveru da dobije vrednost SOA RR-a. Ukoliko sekundarni DNS ne dobije odgovor na svoj upit, ponavljaće svoj upit sa periodom koja je zadata kao *retry* tajmer u SOA RR-u. Ako ne stigne odgovor od primarnog DNS servera u vremenu definisanom sa *expire* tajmerom u okviru SOA RR-a, sekundarni DNS server će da obriše svoje podatke za navedeni domen. Posle brisanja podataka o domenu za koji je bio sekundarni DNS, na odgovore na upit vezane za ovaj domen, ponašaće se kao bilo koji drugi DNS server koji nema podataka o domenu za koji je stigao upit.

U prethodnom scenariju navedeno je da administratori menjaju podatke u zonskom fajlu za neki domen na primarnom DNS serveru. Uvođenjem DHCP-a u širu upotrebu, ručna izmena zonskog fajla više nije bila adekvatna. Umesto toga, definisan je protokol za dinamičku izmenu podataka u zonskom fajlu. U okviru RFC 2136 (*Dynamic Updates in the*

Domain Name System) definisan je mehanizam pomoću kog DHCP server može da doda ili obriše RR i/ili RR-set.

Sa stanovišta odgovaranja na upite, primarni i sekundarni DNS serveri su ravnopravni, tj. svi navedeni serveri ravnopravno odgovaraju na upite koji dolaze sa Interneta. Svaki primarni i sekundarni DNS server je ujedno i autoritativni DNS server za dati domen. Svi ostali DNS serveri, koje nazivamo rekurzivni DNS serveri, koji dobijaju podatke od autoritativnih DNS servera pa ih dalje prosleđuju klijentima, su neautoritativni DNS serveri.

Za primarni i sekundarne DNS servere je bitno napomenuti da ne treba da se nalaze na istoj logičkoj mreži. Ovo ograničenje je uvedeno sa idejom da komunikacioni problemi na jednoj mreži ne dovedu do potpunog gubitka sa Interneta DNS servera za jedan domen što bi automatski učinilo ne vidljivim sve hostove iz navedenog domena. Ovo ograničenje nije ugrađeno u sam DNS protokol već je u pitanju preporuka koje su dužni da se pridržavaju svi vlasnici domena.

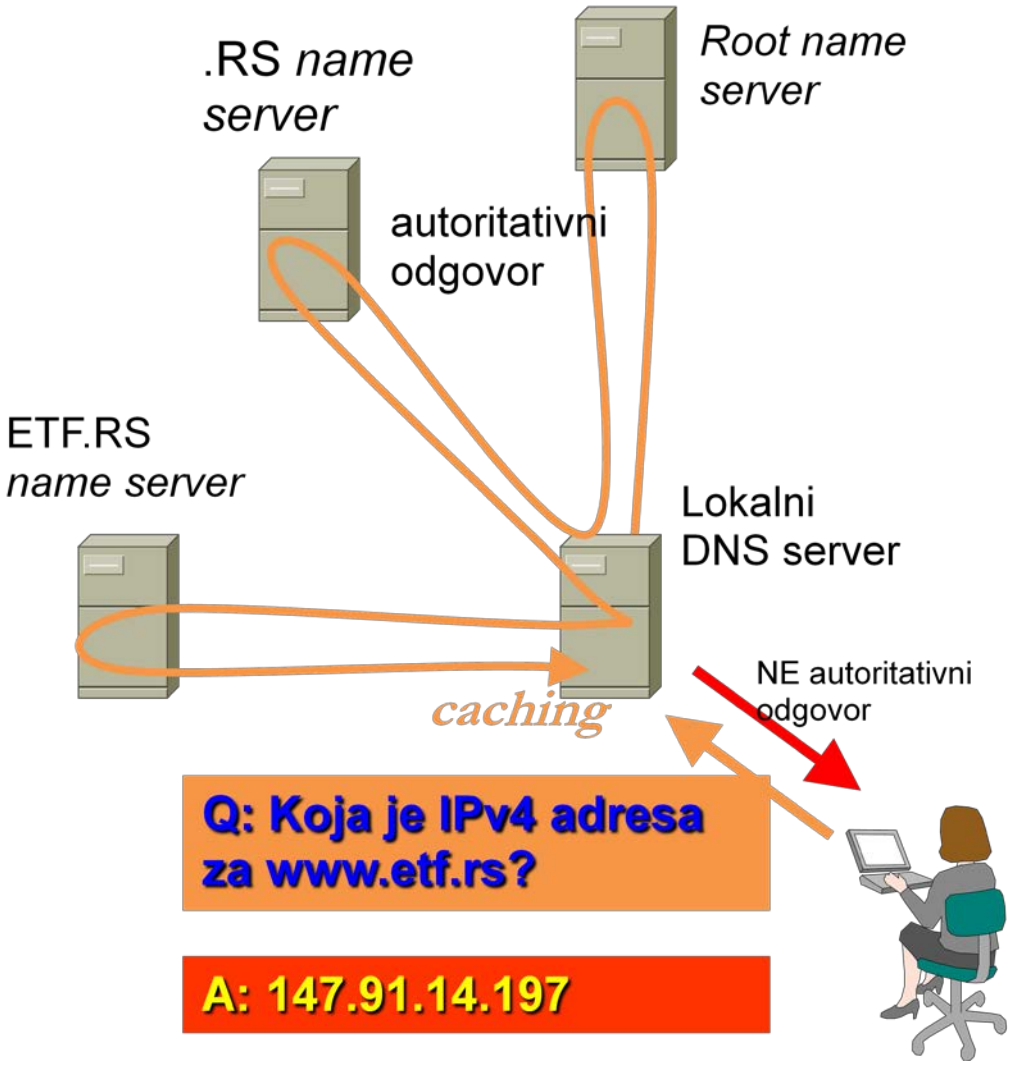
4.2 Slanje upita i dobijanje odgovora

U svakodnevnom radu na Internetu, korisničke aplikacije, po pravilu, pre svakog pristupa Internet servisu, šalju DNS upit za dobijanje IP adrese računara sa kojim treba da uspostave komunikaciju. Tipičan primer tog procesa predstavljen je na slici 2. Upiti koji se šalju mogu da budu iterativni ili rekurzivni. Kod iterativnog upita, DNS server vraća informaciju koju trenutno ima kod sebe u vezi upita (na primeru na slici 2, upiti koje šalje lokalni DNS server su iterativni). Kod rekurzivnog upita, klijent koji je poslao upit očekuje od DNS servera da primenom rekurzivnog procesa u potpunosti odredi informaciju koja je tražena i da je dostavi klijentu.

Na korisničkom računaru, po iniciranju uspostave komunikacije od strane korisnika, aplikacija od DNS *resolver*-a na računaru zahteva dobijanje IP adrese. DNS *resolver* na računaru šalje DNS upit svom lokalnom DNS serveru. IP adresa DNS servera se postavlja prilikom inicijalnog konfigurisanja za povezivanje na mrežu ili se automatski dobija posredstvom DHCP (*Dynamic Host Configuration Protocol* - RFC 2131) protokola. Po pravilu, klijentski računar se uvek obraća rekurzivnom DNS serveru šaljući rekurzivni upit. Zadatak ovog servera je da, ukoliko nije autoritativni DNS server za domen na koji se odnosi upit, za potrebe klijenta pronađe IP adresu u okviru DNS sistema. Na primeru sa slike 2 klijent se obratio svom rekurzivnom DNS serveru tražeći da dobije IP adresu sajta *www.etf.rs*. Pošto se radi o rekurzivnom DNS serveru koji nije autoritativni server za domen *etf.rs*, on je poslao upit *root* DNS serveru. *Root* DNS server nema podatke o hostu *www.etf.rs* ali zna koji su autoritativni DNS serveri za domen *.rs* i te podatke vraća rekurzivnom DNS serveru.

U drugom koraku, rekurzivni DNS server šalje upit u kome traži IP adresu za ime *www.etf.rs* ka DNS serveru nadležnom za *.rs* domen. Ovaj DNS server nije autoritativni server za domen *etf.rs*, ali zna koji su DNS serveri autoritativni za taj domen i tu informaciju i vraća rekurzivnom serveru. Rekurzivni server ponovo šalje upit, ali ovaj put DNS serveru koji je autoritativni DNS server za domen *etf.rs*. Autoritativni DNS server za domen *etf.rs* će vratiti u svom odgovoru IP adresu za host *www.etf.rs*. Takođe će vratiti i listu autoritativnih DNS servera za domen *etf.rs*. Rekurzivni DNS server će po dobijanju odgovora od

autoritativnog DNS servera da tu informaciju upiše u posebnu memoriju za keširanje podataka i da ga zatim prosledi klijentu koji je poslao upit. Keširanje podataka na rekurzivnom DNS serveru se vrši u cilju ubrzavanja procesa dobijanja IP adrese na osnovu imena. Sledeći klijent koji bude postavio isto pitanje pre nego što istekne dozvoljeno vreme keširanja podataka (TTL - *Time To Live*) dobiće informacije direktno od svog rekurzivnog DNS servera, bez potrebe da se traži autoritativni DNS server. Pored toga što rekurzivni DNS server kešira podatke dobijene od autoritativnih DNS servera, i sam klijent će neko vreme čuvati navedene podatke.



Slika 2 - Primer dobijanja IPv4 adrese

Prilikom slanja upita DNS serveru, klijent (*resolver*) navodi koji tip RR-a želi da dobije za naziv domena/hosta koji je naveo. Ukoliko želimo da odredimo IPv4 adresu, u pitanju je A tip RR-a. Kada su u pitanju adrese web sajtova, vrlo često su nazivi tipa *www.domen.tld* dati u formi alternativnog naziva za neki drugi host, tj. u formi CNAME (*Canonical NAME*) RR-a. U takvim situacijama, iako je u upitu tražen A tip RR-a, autoritativni DNS server će vratiti takozvani RR-set koji se sastoji od CNAME RR-a i A RR-a za host na koji ukazuje CNAME RR.

4.3 Struktura zonskih fajlova i RR-ova

Kada govorimo o nekom domenu, prvenstveno mislimo na sve one podatke koji se vezuju za taj domen a nalaze se u zonskim fajlovima. Kao što je ranije rečeno, svi podaci u DNS prostoru koji se odnose na neki domen nalaze se u određenom zonskom fajlu. Zonski fajl se sastoji od *Resource Record*-a (RR). Standardna struktura RR-a data je u formi:

labela1.labela2.labela3. *TTL* **IN** *tip_RR* *podatak*

gde je: *labela1.labela2.labela3.* - naziv poddomena ili hosta koji se sastoji iz jedne ili više labela razdvojenih tačkom; prema RFC-u 952, labele se sastoje od slova engleskog alfabeta i brojeva - [A...Z,0...9] i ne pravi se razlika između velikih i malih slova,
TTL - Time To Live - vreme koje definiše koliko dugo podatak sme da se čuva na *caching* serveru,
IN - klasa RR-a. Uvek je IN
tip_RR - Ovde se navodi koji je tip RR-a.
podatak - sam podatak koji se vezuje za naziv poddomena ili hosta.

Postoji više tipova RR-ova koji su definisani u RFC-u 1035:

- *Terminal* RR
  - *Address Records*
    - A - IPv4 adresa
    - AAAA - IPv6 adresa
  - *Informational/Data* (TXT, HINFO, KEY, SSHFP)
- *Non-terminal* RR (MX, SRV, PTR, KX, A6, NAPTR, AFSDB)
- *Indirect* RR (CNAME, DNAME)
- DNS *Internal Types* (SOA, NS, DS, DNSKEY, RRSIG, NSEC, NSEC3)

Tipična struktura zonskog fajla je sledeća:

```
; ETF.BG.AC.RS definicija domena
$TTL 3h
@       IN      SOA     NS.ETF.BG.AC.RS HOSTMASTER.ETF.BG.AC.RS (
                        2015050600 ; serial
                        3h ; refresh
                        1h ; retry
                        1w ; expire
                        2h) ; negative caching
; DNS serveri za domen
                IN      NS      ns.etf.bg.ac.rs.
                IN      NS      zmaj.etf.bg.ac.rs.
                IN      NS      odisej.telekom.rs.
; Mail serveri za domen
                IN      MX      10      smtp.etf.bg.ac.rs.
                IN      MX      15      smtp2.etf.bg.ac.rs.
; web sajt
                IN      A       147.91.14.197
www             IN      A       147.91.14.197
```

```
; poddomeni
ep              IN      NS      ns.etf.bg.ac.rs.
                IN      NS      zmaj.etf.bg.ac.rs.
; ostali podaci koji se odnose na domen
svarog          IN      A       147.91.8.238
smtp            IN      A       147.91.14.160
smtp2           IN      A       147.91.13.5
```

Svaki zonski fajl počinje sa direktivom "TTL" kojom se definiše, za sve RR-ove u zonskom fajlu, koliko će dugo podaci moći da se čuvaju u kešu *caching* DNS servera. Sledeći RR, koji je obavezan, je SOA RR. SOA RR definiše serijski broj i tajmere koji su relevantni za rad DNS servisa. Ostatak zonskog fajla čine različiti RR-ovi koji opisuju podatke vezane za navedeni domen.

4.4 DNS *Security* (DNSSEC)

Veliki problem u funkcionisanju DNS servisa predstavlja sigurnost. DNS servis je nastao u doba kada su na Internet dominantno bile povezane naučno-istraživačke organizacije. U takvom okruženju bio je relativno mali broj onih koji bi ugrozili sigurnost i stabilnost rada Interneta. Komercijalizacijom Interneta stvari su se iz osnova promenile. Danas kada kažemo da je Internet mreža svih mreža, kažemo i da su na Internetu i razni korisnici kojima je jedini cilj zloupotreba Interneta i kompromitacija rada. U takvoj sredini dolaze do izražaja sve slabosti protokola i servisa koji u svom nastajanju tom pitanju nisu posvetili dovoljno pažnje. Jedan od takvih servisa je i DNS servis. Vremenom se pokazalo da DNS servis uopšte ne spada u sigurne servise. Korisnik koji je primio DNS odgovor ne može da zna odakle je odgovor stvarno došao, da li su podaci koje je server poslao tačni i da li je korisnik primio podatke koje je server poslao. Ovako predstavljene činjenice mogu da navedu korisnika da pomisli da ne treba da koristi DNS servis uopšte. Pošto ne postoji alternativni servis za DNS, jedino što je ostalo je da se povećanje sigurnost DNS servisa. To je urađeno uvođenjem *Secure* DNS-a ili skraćeno DNSSEC-a. Osnovna novina koju je doneo DNSSEC je digitalno potpisivanje podataka koji se razmenjuju putem DNS protokola. Uvođenjem digitalnog potpisivanja podataka, sprečava se modifikacija podataka u transportu i omogućava se provera ko je stvarno poslao te podatke. Da bi sistem funkcionisao, potrebno je bilo uvesti lanac poverenja koji polazi od *root* DNS servera i proteže se dalje duž stabla naziva domena. Da bi ovaj lanac poverenja funkcionisao, neophodno je da budu potpisani svi domeni između *root* domena i našeg domena. Na primer, da bi mogao da se uvede DNSSEC mehanizam digitalnog potpisivanja, na odgovarajući način, za domen etf.bg.ac.rs, neophodno je da digitalno potpisivanje bude uvedeno za .rs, ac.rs i bg.ac.rs domen. Tek tada i domen etf.bg.ac.rs može da uvede digitalno potpisivanje svojih podataka. S obzirom da digitalno potpisivanje DNS podataka još nije uvedeno za .rs domen, nije ga moguće uvesti ni za ostale domene ispod .rs *top-level* domena.

Treba imati u vidu da je digitalno potpisivanje DNS podataka samo polovina rešenja problema loše sigurnosti DNS servisa. Da bi sigurnost DNS servisa bila potpuna, neophodno je da i svi rekurzivni i *resolver*-ski DNS serveri uvedu proveru digitalnog potpisa za podatke koje dobijaju od ostalih DNS servera. Za uvođenje provere digitalnog potpisa DNS RR podataka nije potrebno da vaš domen bude potpisan. Ova funkcionalnost može da se uvede na svakom rekurzivnom DNS serveru nezavisno od svega ostalog, i to je nešto što se danas

preporučuje svima koji imaju svoje rekurzivne DNS servere. Uvođenjem DNSSEC provere na rekurzivnim DNS serverima sprečavaju se modifikacija DNS podataka u transportu kao i poturanje lažnih DNS podataka u slučaju kada su ti DNS podaci digitalno potpisani u skladu sa pravilima koja su definisana u okviru DNSSEC-a. Trenutno važeća specifikacija DNSSEC-a data je u RFC-ovima 4033, 4034 i 4035. Zahvaljući tome, *root* zona je digitalno potpisana 15. jula 2010. godine. Već 29. jula 2010. godine je potpisan .edu domen, pa 9. decembra 2010. godine .net domen i, na kraju, 31. marta 2011. godine .com domen. To je otvorilo mogućnost da i korisnici Internet domena mogu da počnu sa primenom DNSSEC-a i u svojim domenima.

Da bi DNSSEC mogao normalno da funkcioniše, neophodno je da DNS server podržava ekstenziju DNS protokola poznatiju kao EDNSO (RFC 6891) koja dozvoljava da UDP paket ima dužinu do 4kB. Pored podrške za EDNSO na samom DNS serveru, potrebno je proveriti da li mrežni *firewall* uređaj podržava EDNSO. Ukoliko *firewall* podržava samo standardne UDP pakete sa DNS sadržajem dužine do 512 bajtova, svi duži UDP paketi će biti identifikovani kao neispravni i kao takvi će biti odbačeni.

Još jedan preduslov da bi mogao da se koristi je da vreme na DNS serveru bude sinhronizovano na neki od referentnih izvora vremena. Ukoliko to nije ispunjeno, pošto se kod DNSSEC potpisivanja i validacije koristi i vremenska referenca, ove operacije neće proći. Zbog toga je važno obezbediti pouzdano korišćenje NTP protokola ili instalirati GPS prijemnik sa odgovarajućim softverom za sinhronizaciju tačnog vremena.

5 Napadi na DNS

Imajući u vidu značaj DNS servisa za funkcionisanje Interneta, sve je veći broj različitih napada koji najčešće za cilj imaju kompromitovanje DNS podataka, a neki put i potpunu blokadu rada (*Denial of Service*). Iako je lista mogućih napada na DNS dosta dugačka, ovde će biti obrađeni neki tipični napadi koji su najčešći u praksi.

Nedostaci u implementaciji DNS servisa mogu da se iskoriste za zlonamerne aktivnosti. Kako je DNS kritični protokol za Internet poslovanje, bezbrojne operativne sisteme i aplikacije, administratori moraju da povećaju zaštitu DNS servera kako bi sprečili da se zloupotrebe. U nastavku su prikazani neki od nedostataka u implementaciji DNS protokola, kao i tehnike pomoću kojih možemo sprečiti da se ti nedostaci zloupotrebe.

5.1 Napadi na server na kome se hostuje DNS server

Kao i svaki Internet servis, tako je i DNS servis pogođen svim vrstama napada koji pogađaju platformu na kojoj se hostuje DNS servis. Ovde navodimo najčešće probleme na DNS serverima koji mogu da kompromituju rad DNS servisa:

- *Bug* u operativnom sistemu ili bilo kojoj drugoj aplikaciji koja se nalazi na serveru koji hostuje DNS servis.
- *Packet flooding* napad na TCP/IP komunikacioni stek na DNS serveru koji će onemogućiti slanje/prijem paketa.
- ARP *spoofing* napad od strane insajdera koji se nalazi na istom VLAN-u/LAN-u na kom i DNS server.
- Virusi i "crvi" koji mogu da kompromituju sadržaje fajlova relevatnih za rad DNS servisa, poput *named.conf, root.hints,* zonskih fajlova, *resolv.conf, host.conf*.
- Kompromitovan host na istom VLAN-u/LAN-u na kome se nalazi i DNS server, može da presretne DNS saobraćaj i da modifikuje odgovore čime bi saobraćaj preusmerio na lažne DNS servere.

Pored prethodno navedenih problema koji se odnose na platformu na kojoj se hostuje DNS servis, određen broj problema može da potiče i od same softverske implementacije DNS servisa. Tipični problemi sa implementacijom DNS servisa se odnose na bagove poput *buffer overflow* napada koji tipično dovodi do prekida rada DNS servisa, što spada u grupu *Denial of Service* (DoS) napada.

5.2 Greške u konfiguraciji DNS servisa

Pored spoljašnjih napada koji mogu dovesti do prestanka rada DNS servisa, sličan efekat se dobija i u situacijama kada primarni i sekundarni DNS serveri nisu ispravno konfigurisani. Tipične greške ovog tipa su:

- *Lame Delegation* - Kada se desi da u *child* zoni dođe do izmene IP adresa DNS servera ali se ta izmena ne uradi i u *parent* zoni, dolazi do nedostupnosti *child* zone.

- *Zone drift* - Ukoliko su vrednosti tajmera za *Refresh* i *Retry* postavljeni na suviše velike vrednosti, a sam zonski fajl se često menja, može doći do gubitka sinhronizacije između sadržaja zona na primarnom i sekundarnim serverima. Ovu neusaglašenost nazivamo *zone drift*.
- *Zone thrash* - Ukoliko su vrednosti tajmera za *Refresh* i *Retry* postavljeni na suviše malu vrednost, sekundarni serveri će suviše često proveravati SOA RR na primarnom DNS serveru čime donekle povećavaju opterećenje hardvera primarnog DNS servera. U zavisnosti od snage hardvera na primarnom DNS serveru, u krajnjem slučaju može doći i do DoS simptoma.
- Informacije u okviru RR HINFO i TXT sa podacima o verzijama hardvera i softvera na DNS serveru mogu da pruže potencijalnom napadaču dragocene informacije na osnovu kojih će doći do poznatih bagova za navedeni hardver i softver čime mu se otvaraju vrata za izvršenje napada.

5.3 DNS *Open resolvers*

DNS *open resolver* je DNS server koji dozvoljava klijentima koji nisu deo njegovog administrativnog domena da koriste taj server za rekurzivni upit. U suštini, DNS *open resolver* daje odgovore na upite koji mogu poticati od bilo kojih korisnika. Zbog toga, DNS *open resolver*-i su osetljivi na više zlonamernih aktivnosti, uključujući sledeće:

- DNS *cache poisoning* napade,
- *Resource utilization* napade,
- DNS *Amplification and Reflection*
- *Denial of Service* (DoS), *Distributed* DoS (DDoS).

5.3.1 DNS *Cache Poisoning Attacks*

DNS cache poisoning napad nastaje kada napadač šalje falsifikovane i obično lažne zapise *resolver*-u. Kada *resolver* prihvati lažne informacije, one se čuvaju u DNS kešu trajno, odnosno do isteka TTL-a *(Time to Live).* Da bi zloupotrebio ovu osobinu DNS *resolver*-a, napadač mora biti u stanju da pravilno predvidi identifikator DNS transakcija (TXID) i UDP izvorišni port DNS upita. Ukoliko *DNS* server nije ispravno konfigurisan da koristi slučajne vrednosti za izvorišni UDP port već stalno koristi port 53 (nekada je to bilo uobičajeno), tada je napadaču olakšan napad na *DNS* server. Napadači koriste ovu tehniku da preusmere korisnike sa legitimnih sajtova ka kopijama originalnih sajtova koje su kompromitovane, ili da obaveste *resolver* da koristi kompromitovani *name* server (NS) sa lažnim zapisima koji će se koristi za zlonamerne aktivnosti.

5.3.2 *Resource Utilization Attacks*

Resource utilization napadi na DNS *open resolver*-e troše resurse na uređajima na kojima je *resolver*, što uključuje procesor, memoriju i *socket* bafere. Ovim vrstama napada pokušava se da se potroše svi raspoloživi resursi kako bi se onemogućio rad *resolver*-a. Kao posledica ovih napada obično se mora zaustaviti i restartovati DNS servis ili restartovati kompletan server.

5.3.3 DNS *Amplification and Reflection Attacks*

DNS *Amplification and Reflection* napadi iskorišćavaju DNS *open resolver*-e sa ciljem da se poveća obim napada i da se sakrije pravi izvor napada, što obično rezultira DoS ili DDoS napadom. Ovi napadi su mogući jer *open resolver*-i odgovoraju na upite svakome ko pošalje upit. Napadači koriste te DNS *open resolver*-e za zlonamerne aktivnosti slanjem upita DNS *resolver*-ima koristeći falsifikovane izvorišne IP adrese koje su meta napada. Kada *resolver*-i dobiju DNS upite sa lažiranom izvornišnom IP adresom, oni odgovaraju slanjem odgovora na odredišnu adresu koja je meta napada. Napadi ove vrste obično koriste veći broj DNS *open resolver*-a tako da efekti budu uvećani na krajnjim uređajima koji su meta napada.

5.4 *Maliciously Abusing Resource Record Time To Live*

Kada DNS *resolver* pošalje upit tražeći informacije, autoritativni ili ne-autoritativni server može odgovoriti sa relevantnim RR *(resource record)* podatakom ili greškom. RR podatak sadrži 32-bitno TTL *(Time to Live)* polje preko koga se obaveštava *resolver* koliko dugo može biti u kešu sačuvan RR podatak, nakon čega *resolver* treba ponovo da pošalje DNS upit. TTL polje se može iskoristiti zlonamerno, postavljanjem na veoma malu ili veliku vrednost. Pomoću male TTL vrednosti, napadači mogu da iskoriste DNS da distribuiraju zlonamerne informacije DNS *resolver*-ima o velikom broju kompromitovanih uređaja. Mapiranje imena u IP adrese za uređaje u traženom domenu brzo će se menjati (obično u rasponu od nekoliko sekundi do nekoliko minuta), što je poznato kao *Fast-Flux* (FF) mreža. Zloupotreba TTL vrednosti primenjujući ovu tehniku na RR podatake u odgovoru na DNS upit je poznat kao *Single-Flux*. Ova zlonamerna tehnika otežava operaterima da koristeći *traceback* metode identifikuju kompromitovane uređaje koji učestvuju u *Fast-Flux* mreži.

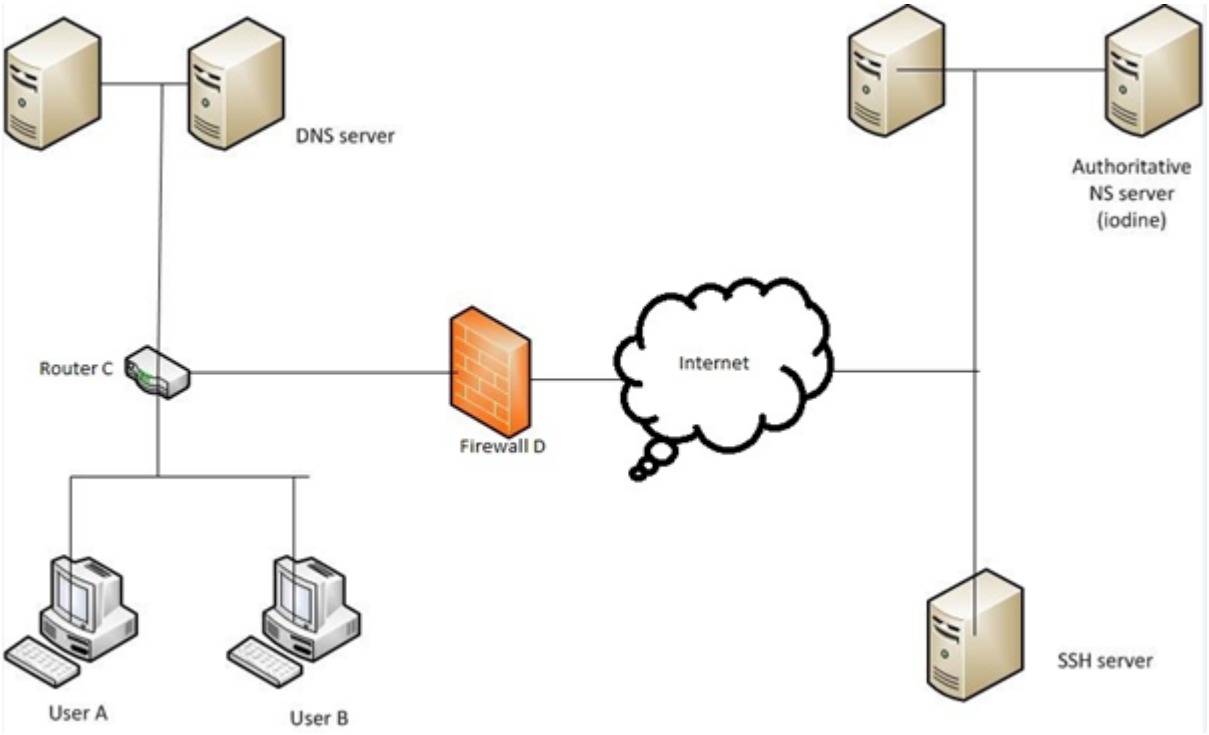
Napadači mogu da koriste i višestruku tehniku, odnosno, da brzo menjaju mapiranje imena u IP adrese za oba DNS zapisa i A i NS, stvarajući *Double-Flux* (DF) mrežu. Još jedna potencijalno zlonamerna upotreba male vrednosti TTL polja je kada je vrednost 0. U tom slučaju DNS *resolver* dobija informacije da RR podatak koji je primio u odgovoru uopšte ne treba da sačuva u kešu.

Napadači mogu da koriste velike vrednosti TTL polja tako što će DNS *resolver* da kešira na duži vremenski period informacije primljene u odgovoru na upit. Ova tehnika se može koristiti za čuvanje zlonamernih RR podataka u kešu jednog *resolver*-a za duži vremenski period. Ako je *resolver* rekurzivni ili *open resolver*, onda on može distribuirati RR podatke malicioznih uređaja mnogim *resolver*-ima, što se može iskoristiti za zlonamerne aktivnosti. Ova metoda se razlikuje od *Fast-Flux* tehnike koja koristi malu TTL vrednost i operateri su u stanju da korišćenjem *traceback* tehnike lakše identifikuju maliciozne uređaje koji vrše distribuciju te informacije.

5.5 DNS *tunneling*

U situacijama u kojima ne može da se pristupi nekom veb sajtu, jer je blokiran preko *proxy* servera, može se koristiti DNS *tunnelling* da se zaobiđe *proxy* server. Korišćenjem DNS tunelovanja, korisnik će biti u prilici da pristupi veb sajtu iako ga *proxy* server blokira. Kad se koristi *proxy* server, sav HTTP saobraćaj će biti prvo primljen od strane *proxy* servera, dok će DNS saobraćaj prolaziti. Eksploatacijom tog DNS saobraćaja omogućeno je da se koriste svi blokirani sajtovi od starne *proxy* servera.

Dakle, korišćenjem DNS tunela, podaci se enkapsuliraju u okviru DNS upita i odgovora, korišćenjem *base32* i *base64* enkodovanja i sa DNS *Domain Name Lookup* sistemom se šalju podaci dvosmerno. Dakle, dokle god može da se uradi upit za ime domena na mreži, mogu se tunelovati bilo kakvi podaci do udaljenog sistema, uključujući i Internet. Da se enkapsulira IP saobraćaj koriste se sledeći DNS zapisi null, TXT, SRV, MX, CNAME.



Slika 3 - Primer DNS *tunneling*-a

U navedenom scenariju (slika 3), korisnici A i B se nalaze iza korporativnog *firewall* uređaja D i nije im dozvoljen web saobraćaj, već je dozvoljen od *firewall*-a samo saobraćaj preko porta 53. Korisnici A i B mogu da pristupe Internetu eksploatacijom DNS saobraćaja pomoću DNS tunela. DNS server na levoj strani ima sposobnost da kešira podake, tako da kada korisnik pokuša da pristupi nekim web sajtovima koji su već u kešu, onda zahtev neće ići ka iterativnom serveru. Ako stigne novi zahtev od korisnika A, pri čemu nema A zapisa u DNS serveru, on će proslediti upit ka spoljnom DNS serveru. Maksimalna dužina DNS upita je 255 znakova sa ograničenjem od 63 karaktera po labeli (label3.label2.label1.example).

Da bi tunelovali podatke preko DNS-a moramo imati kontrolu nad spoljnim DNS serverom (u našem slučaju, DNS server sa desne strane), na kome treba da dodamo dva zapisa. Jedan je NS zapis a drugi A zapis. NS (*name* server) zapis nam omogućava da delegiramo poddomen našeg domena na drugi server. Dakle, ako imamo domen ***laptop.com*** možemo dodati NS zapis kao ***a.laptop.com NS computer.com*** što znači bilo kakav DNS upit ka ***laptop.com*** će biti delegiran ka ***computer.com*** i njegovim poddomenima. Drugi je A zapis, koji sadrži IP adresu mapiranu sa imenom domena.

Za podešavanje servera za DNS tunelovanje, mogu se instalirati *Iodine, DNS tunnelling script, DNScapy* i drugi alati. Dakle, na krajnjoj mašini, mora da postoji klijent za DNS tunelovanje. Kada se jednom uspostavi tunel, može se koristiti SOCKS *proxy* za

nesmetanu vezu. DNS tunelovanje je neefikasano i sporo, jer DNS saobraćaj ima ograničen protok za podatke pošto ima samo sposobnost da prenosi male količine informacija kao što su DNS upit i odgovor.

Bez obzira na loše osobine rešenja na bazi DNS tunelovanja, napadači za *botnet* mogu da koriste DNS tunelovanja kao skriveni kanal za komunikaciju koji je veoma teško otkriti. Pošto se za komunikaciju u okviru *botnet*-a koristi mali kapacitet, ograničenje DNS tunelovanja po pitanju protoka u ovom slučaju ne predstavlja problem. Postoji veliki broj softverskih alatki koje omogućavaju formiranje i korišćenje DNS tunela:

DNS TUNNELLING TOOLS
OzymanDNS
Dns2tcp
Iodine
Heyoka
DNSCat
NSTX
DNScapy
MagicTunnel, Element53, VPN-over-DNS (Android)
VPN over DNS

U DNS tunelovanju, zahtevi od klijenata se dele na fragmente i šalju kao odvojeni DNS upiti, pa isto tako i odgovor mora da se deli na fragmente. DNS koristi UDP umesto TCP protokola, tako da fragmentacija i ispravno sastavljanje fragmenata treba da se uradi na lažnom serveru.

5.5.1 Otkrivanje DNS *tunneling*-a

- DNS tunel može se otkriti praćenjem veličine DNS zahteva i odgovora na zahteve. Najverovatnije je da će DNS paketi koji se koriste za tunelovanje imati više od 64 znakova.
- Korišćenje ažuriranih IPS i IDS sistema je još jedan mehanizam za detekciju
- *Firewall*, IDS i IPS moraju da budu konfigurisani da prate ukoliko se pojavi veliki broj TXT zapisa u DNS-u.
- Konfigurisana pravila u SIEM *(Security information and event management)* koja se trigeruju ako je obim DNS saobraćaja od određenog izvora veoma visok.
- Još jedan metod je korišćenje *split horizon* DNS koncepta, tako da se interne adrese pojavljuju na unutrašnjem DNS serveru. Klijenti treba da koriste *proxy* server za povezivanje na internet koji će da rezrešava DNS upite ka spolja za njih.
- Pomoću DNSTrap alata koji je razvijen da otkrije DNS tunelovanje pomoću veštačke neuronske mreže. U njemu se koriste pet atributa da se obuče neuronske mreže (*Artificial Neural Network* ANN) za detekciju tunela. To su naziv domena, broj paketa koji se šalju

određenom domenu, prosečne dužine paketa u tom domenu, prosečni broj različitih karaktera u LLD *(Lowest Level Domain)* i udaljenost između LLD-ova.

- Pomoću *firewall* uređaja novije generacije, koji imaju sposobnost da detektuju DNS tunele.

6 Preporuke za konfigurisanje DNS servera

Zaštita DNS servera je delikatan posao. Pored pravilnog izbora operativnog sistema za hosting DNS servisa, potrebno je obratiti pažnju i na izbor softvera za sam DNS servis. Imajući to u vidu, prilikom izbora platforme za DNS servis, potrebno je obratiti pažnju na sledeće:

- Treba koristiti najnoviju stabilnu verziju softvera za DNS servis ili pretposlednju verziju softvera sa primenjenim svim patch-evima.
- Pokretanje DNS softvera sa privilegijama sa restrikcijom.
- Razdvojiti autoritativne DNS server od neautoritativnih DNS servera.
- Hostovati DNS server na serveru koji je posvećen samo toj funkciji.
- Obezbediti topološku i geografsku raspodelu primarnih DNS servera po celom svetu.

6.1 *Randomization for* DNS *Transaction Identifier*

Za praćenje upita i odgovora na upite DNS koristi identifikatore transakcija (TXID). DNS TXID je 16-bitno polje u zaglavlju DNS poruke, koje DNS koristi zajedno sa vrednošću izvorišnog porta kako bi sinhronizovao odgovore sa prethodno poslatim upitima. Na DNS-ovima gde ne postoji dovoljna entropija u nasumičnom raspoređivanju DNS identifikatora transakcija prilikom upita, moguće je izvršiti zloupotrebe. Napadač može analizirati TXID vrednosti koje generiše DNS i na osnovu njih kreirati algoritam koji se može iskoristiti za predviđanje sledeće TXID vrednosti za naredne upite. Ako je napadač u stanju da predvidi sledeći broj identifikatora transakcije koji se koristi prilikom DNS upita, zajedno sa vrednošću izvorišnog porta, može da konstruiše i pošalje kompromitovani DNS odgovor sa ispravnom vrednošću identifikatora transakcije. Iako je DNS poruka poslata od strane napadača falsifikovana, *resolver* prihvata odgovor jer se vrednost identifikatora transakcije i odredišnog porta podudaraju sa upitom koji je *resolver* poslao, što rezultira sa kompromitovanim kešom *resolver*-a. TXID polje za DNS protokol je dužine 16 bita, tako da ova vrednost može da varira od 0 do 65535.

Tokom konfiguracije BIND-a za UNIX i LINUX sisteme, preporučuje se da operateri koriste */dev/random* sa *--with-randomdev=PATH* argumentom u konfiguracinoj skripti. */dev/random* je poseban fajl koji se koristi za generisanje slučajnih brojeva, poznat kao generator slučajnih brojeva „*random number generator* (RNG)“ ili pseudoslučajni generator brojeva (PRNG). Na ostalim operativnim sistemima generator slučajnih brojeva se drugačije implementira. */dev/random* se preporučuje jer kreira entropijski *pool* (grupu slučajnih bita koji se nalaze na jednom mestu) za generisanje nepredvidljivih slučajnih brojeva. Kada su svi biti iscrpljeni iz entropijskog *pool*-a, kreira se novi *pool* koji sadrži slučajne bite. Korišćenje */dev/random* generatora slučajnih brojeva omogućava BIND-u da generiše nasumične identifikatore transakcija (TXID).

Za korišćenje generatora slučajnih brojeva (*/dev/random*) kada se konfiguriše BIND potrebno je zadati sledeće:

[user@server ~/bind-9.5.0]$ ./configure --with-randomdev=/dev/random

Za Microsoft DNS server treba ispoštovati preporuke proizvođača:
https://technet.microsoft.com/en-us/library/ee649266%28v=ws.10%29.aspx

6.2 UDP *Source Port Randomization*

Za praćenje upita i odgovora na upite DNS koristi vrednost izvorišnog porta i identifikatora transakcije. U DNS-ovima gde ne postoji dovoljna entropija u nasumičnom raspoređivanju UDP izvorišnog porta prilikom upita, moguće je izvršiti zloupotrebe . Napadač može analizirati vrednosti UDP izvorišnog porta koje generiše DNS i na osnovu njih kreirati algoritam koji se može iskoristiti za predviđanje sledeće vrednosti UDP izvorišnog porta za naredne upite. Ako je napadač u stanju da predvidi sledeću vrednost UDP izvorišnog porta kao i vrednost identifikatora transakcije koji se koristi prilikom DNS upita, može da konstruiše i pošalje kompromitovani DNS odgovor sa ispravnom vrednošću UDP izvorišnog porta. Iako je DNS poruka poslata od strane napadača falsifikovana, *resolver* prihvata odgovor jer se vrednost UDP izvorišnog porta i identifikatora transakcije podudaraju sa upitom koji je *resolver* poslao, što rezultira sa kompromitovanim kešom *resolver*-a. Polje izvorišnog porta za UDP protokol je dužine 16 bita, tako da ova vrednost može da varira od 0 do 65535.

6.3 Odvajanje autoritativnih i rekurzivnih *resolver* DNS servera

Autoritativni i rekurzivni *resolver*-i imaju različite primarne funkcije. Autoritativni DNS server distribuira informacije DNS *resolver*-ima za imena u okviru svog administrativnog domena. Rekurzivan *resolver* rekurzivno prolazi kroz DNS arhitekturu i locira autoritativan DNS server nadležan za informacije iz DNS upita, a zatim distribuira odgovor ili grešku za te informacije *resolver*-u koji je postavio pitanje. Pošto se funkcije ovih *resolver*-a koriste za različite svrhe, *resolver*-i treba da se odvoje.

Autoritativni DNS serveri treba da se koriste samo za odgovore na upite za imena domena za koje je server nadležan. Upiti od bilo koga (upiti sa Interneta) mogu se odobriti za informacije koje znamo (autoritativne RR podatke).

Rekurzivni DNS serveri treba da se koristi samo za odgovore na upite od DNS *resolver*-a unutar svog administrativnog domena. Upiti poznatih izvora (klijenata unutar svog administrativnog domena) mogu se odobriti za informacije koje ne znamo (na primer, za domenska imena izvan našeg administrativnog domena).

Ova dva tipa DNS servera treba razdvojiti jer u situaciji kada su ove dve funkcije na istom DNS serveru, odgovori dobijeni kroz rekurzivne upite mogu da kompromituju keš na DNS serveru što će dovesti do distribucije pogrešnih podataka.

6.4 Zaštita od *Spoofing*-a

DNS protokol koristi UDP za većinu svojih operacija. Kako je UDP je nekonektivni protokol, i kao takav lako može da se lažira, mnogi od DNS napada se oslanjaju na lažiranje

(spoofing). Da bi se ograničilo lažiranje može se primeniti nekoliko sigurnosnih metoda, opisanih u nastavku.

6.4.1 Unicast revers path forwarding

Unicast Reverse Path Forwarding (*Unicast* RPF) je funkcija koja može da smanji efikasnost paketa sa lažiranim izvorišnim adresama. Mrežni uređaji koriste *Unicast* RPF upoređujući izvorišnu adresu svakog IP paketa sa lokalnom tabelom rutiranja kako bi utvrdili validnost izvorišne adrese. Mada, iako može da detektuje i filtrira neki lažni saobraćaj, *Unicast* RPF ne pruža potpunu zaštitu od *spoofing*-a jer lažni i validni paketi sa istom izvorišnom adresom mogu doći preko istog interfejsa. S druge strane, danas je sve češći slučaj da je rutiranje saobraćaja na Internetu asimetrično. U tom slučaju korišćenje *Unicast* RPF na mrežnim uređajima dovodi do blokade regularnog Internet saobraćaja.

Unicast RPF radi u dva režima: *strict* i *loose*. U strogom režimu (*strict*), funkcija *Unicast* RPF koristi lokalnu tabelu rutiranja da odredi da li je izvorišna adresa paketa dostupna preko interfejsa na kojem je i primljen paket. Ako je dostupna, paket je dozvoljen, a ako nije paket se odbacuje. Strogi režim rada *Unicast* RPF-a je najbolji za implementaciju na granicama mreža gde ne preovladava asimetrični saobraćaj.

U slobodnom režimu rada *(loose) Unicast* RPF-a, ako je izvorišna adresa paketa dostupna preko bilo kog interfejsa na *Unicast* RPF uređaju, paket je dozvoljen. Ako izvorišna adresa IP paketa ne postoji u tabeli rutiranja, paket se odbacuje.

6.4.2 IP *Source Gard*

IP *source guard* je zaštitni mehanizam na drugom sloju koji se oslanja na *Unicast* RPF i DHCP *snooping*, da filtrira lažiran saobraćaj na individualnim portovima mrežnih uređaja. DHCP *snooping*, što je preduslov za IP *source guard*, proverava DHCP saobraćaj unutar VLAN-a da odredi koje su IP adrese dodeljene kojim mrežnim uređajima i na kojim fizičkim portovima se nalaze. Sve te informacije skladišti u DHCP *snooping bindings* tabeli, koju IP *source guard* koristi za filtriranje primljenih IP paketa na mrežnom uređaju. Ako je paket primljen sa izvorišnom adresom koja ne odgovara DHCP *snooping bindings* tabeli, paket se odbacuje.

Implementacija IP *source guard* u okviru pristupnog sloja mreže može efikasno da eliminiše nastanak lažiranog IP saobraćaja. Međutim, kako zahteva da je aktivan i DHCP servis, nije moguće da se IP *source guard* implementira na granicama internih i eksternih mreža.

6.4.3 *Access control lists*

Liste za kontrolu pristupa *(Access Control Lists ACL)* mogu da obezbede zaštitu protiv *spoofing* napada koji koriste neiskorišćeni ili neprovereni adresni prostor. Obično se ove ACL primenjuju na interfejsima na granicama mreže u pravcu dolaznog saobraćaja. *Spoofing* se

može smanjiti u saobraćaju koji potiče iz lokalne mreže primenom ACL-a koje ograničavaju saobraćaj na samo važeće lokalne adrese.

6.5 Zaštita komunikacije između primarnog i sekundarnih DNS servera

Jedan od osetljivih procesa u radu DNS servisa je sinhronizacija između primarnog i sekundarnih DNS servera. U zavisnosti od veličine zonskih fajlova, ova operacija može da bude veoma zahtevna, kako po pitanju performansi primarnog i sekundarnog DNS servera, tako i po pitanju potrebnih komunikacionih kapaciteta. Drugi problem koji može da se javi je kompromitacija sadržaja tokom transporta. Problem zaštite komunikacije tokom procesa ažuriranje rešen je uvođenjem *Transaction SIGnature* (TSIG) mehanizma (RFC 2845). TSIG mehanizam se zasniva na principu digitalnog potpisa. Primenom *hash* funkcije, deljenog tajnog ključa i vremenske reference, vrši se digitalno potpisivanje svakog DNS paketa koji se razmenjuje između dva servera za koje je konfigurisana primena TSIG-a. Korišćenjem TSIG mehanizma sam sadržaj DNS paketa je i dalje vidljiv na mreži ali TSIG garantuje da niko nije menjao sadržaj paketa tokom transporta. Takođe garantuje da je taj DNS paket poslao neko ko zna deljeni tajni ključ. Osnovna slabost TSIG mehanizma je činjenica da se sigurnost bazira na sigurnosti deljenog tajnog ključa. Bilo kakva kompromitacija ovog deljenog tajnog ključa bi ugrozila sigurnost celog DNS servera. Zbog toga se propuručuje da za svaki DNS server sa kojim se komunicira na bazi korišćenja TSIG-a, treba definisati novi tajni ključ.

Pored komunikacije između primarnog i sekundarnih DNS servera, TSIG mehanizam treba koristiti i u slučaju korišćenja dinamičkog *update*-a primarnog DNS servera. Ako se koristi dinamički *update* tada komunikacija između servera koji šalje takav *update* (tipično DHCP server) i primarnog DNS servera treba da bude zaštićena TSIG mehanizmom.

TSIG mehanizam nije pogodan za zaštitu komunikacije klijenata sa DNS serverom jer bi to značilo da klijenti znaju deljeni tajni ključ čime se gubi tajnost samog ključa pa samim tim i efekat korišćenja TSIG mehanizma.

7 Primeri konfiguracija za BIND i Microsoft DNS softver

7.1 Pokretanje BIND softvera sa ograničenim privilegijama

```
chroot -u named -g other -t /var/named
```

gde su: -u - definiše userID sa kojim treba da se pokreće BIND softver;
navedeni userID mora prethodno da bude kreiran;
-g - definiše groupID kome treba da bude dodeljen userID;
-t - definiše direktorijum u kome će se nalaziti BIND softver koji se pokreće.

7.2 Sprečavanje slanja informacije o verziji BIND softvera.

```
options {
    version none;
};
```

7.3 Sprečavanje *Open Resolver* servera

```
1. Disable Recursion
// Disable recursion for the DNS service
//
options {
    allow-query-cache { none; };
    recursion no;
};


2. Permit Recursion from Trusted Sources
// Permit recursive DNS queries for DNS messages with source
// addresses in the 192.168.1.0/24 netblock
//
options {
    allow-recursion {192.168.1.0/24;};
};


3. Permit Queries from Trusted Sources
// Permit DNS queries for DNS messages with source addresses
// in the 192.168.1.0/24 netblock. The 'allow-query-cache'
// options configuration can also be used to limit the IP
// addresses permitted to obtain answers from the cache of
// the DNS server.
//
// Note: The function of 'allow-query-cache' changed between
// BIND version 9.4 and 9.4.1. Additional information about
// using this options configuration can be found in the BIND
// 9.5 Administrator Reference Manual (ARM).
//
options {
    allow-query {192.168.1.0/24;};
};
```

```
4. Permit and Deny Recursion Using an ACL
// Create an Access List (ACL) defined as 'recursive-permit'
// that will permit devices in the ACL to use the DNS server
// for recursive DNS queries.
//
acl recursive-permit {
    192.168.1.0/24; 10.0.1.0/24; 172.16.1.0/24; 172.31.1.0/24;
};

// Create an Access List (ACL) defined as 'rfc5735-deny' that
// will deny devices in the ACL from using the DNS server for
// recursive DNS queries.
//
acl rfc5735-deny {
    0.0.0.0/8; 10.0.0.0/8; 169.254.0.0/16; 172.16.0.0/12;
    192.0.0.0/24; 192.0.2.0/24; 192.88.99.0/24; 192.168.0.0/16;
    198.18.0.0/15; 198.51.100.0/24; 203.0.113.0/24; 224.0.0.0/4;
    240.0.0.0/4;


// Apply 'recursive-permit' ACL created above to the options
// 'allow-query' or 'allow-recursion' configuration and then
// apply the 'rfc5735-deny' ACL created above to the 'blackhole'
// configuration.

options {
    // Output Truncated.

    allow-recursion { recursive-permit; };
    allow-query { recursive-permit; };

    // Output Truncated.

    blackhole { rfc5735-deny; };
};

// The 'blackhole' options configuration can be used to prevent
// the DNS server from accepting queries for IP addresses that
// are explicitly configured or defined in an ACL. This option
// will also prevent the DNS server from using devices defined
// in the ACL for resolving queries. The 'blackhole' option may
// also be used to prevent the DNS server from sending queries
// to known malicious DNS servers.
```

7.4 Sprečavanje rekurzije na autoritativnim DNS serverima

7.4.1 Bind9

Dodati sledeće u delu globalnih opcionih parametara:

```
options {
     allow-query-cache { none; };
     recursion no;
};
```

7.4.2 Microsoft DNS Server

U okviru Microsoft DNS konzole uraditi sledeće:

1. Desni klik na DNS server i izabrati Properties.
2. Kliknuti na Advanced Tab.
3. U delu "*Server options*" izabrati "Disable recursion" i kliknuti OK.

Microsoft Windows obezbeđuje i funkciju koja se zove *DNS Server Secure Cache Against Pollution* koja ignoriše RR zapise u DNS odgovorima primljenih od neautoritativnih servera. Ta funkcija je aktivna na *Windows 2000 Service Pack 3 (SP3)* i *Windows Server 2003*. Korišćenje ove funkcije će proizvoditi više upita poslatih od DNS servera.

7.5 Ograničeno dozvoljena rekurzija za određene klijente

U određenim slučajevima je potrebno da se dozvole i rekurzivni upiti ali samo za određeni broj klijenata iz mreže. Ovo je tipično potrebno unutar kompanijske mreže ili kod Internet Servis Provajdera. U nastavku su dati primeri za scenario gde su dobro poznate i kontrolisane IP adrese sa kojih se šalju upiti DNS serveru.

7.5.1 BIND9

U globalnoj konfiguraciji DNS servera treba dodati sledeće:

```
acl corpnets { 192.168.1.0/24; 192.168.2.0/24; };
options {
  allow-query { any; };
  allow-recursion { corpnets; };
};
```

7.5.2 Microsoft DNS Server

Trenutno nije moguće na Microsoft DNS serveru uvesti ograničenje po IP adresa za klijente kojima bi bilo dozvoljeno da izvršavaju rekurzivne upite. Da bi se simulirala funkcionalnost koje je prethodno opisano za Bind, potrebno je da se instaliraju dva DNS servera. Jedan DNS server bi omogućavao rekurzivne upite a drugi bi se ponašao kao autoritativni server. Mrežni *firewall* treba da blokira dolazni saobraćaj za rekurzivni DNS server.

7.6 Response Rate Limiting (RRL)

Trenutno je dostupna eksperimentalna funkcija, kao set pečeva za bind 9.9.4 a regularno uključena od verzije 9.10, koja omogućava administratoru da ograniči maksimalan broj odgovora u sekundi koji se šalju jednom klijentu sa imenom servera. Ova funkcionalnost je namenjena da se koristi samo na autoritativnom serveru jer bi ona ugrozila performanse rekurzivnog DNS servera. Da bi se obezbedila najefikasnija zaštita, preporučuje se da autoritativni i rekurzivni DNS serveri realizuju na različitim fizičkim serverima, da se RRL realizuje na autoritativnim serverima. Ovo će smanjiti efikasnost DNS pojačanja napada tako što smanjuje količinu saobraćaja koji dolazi iz bilo kojeg pojedinačnog autoritativnog servera dok ne utiče na performanse unutrašnjih rekurzivnih DNS *resolver*-a.

7.6.1 BIND9

Trenutno postoji set pečeva za verzije 9.8 i 9.9 a uključen je u verziju 9.10 BIND-a, koje omogućavaju RRL na UNIX sistemima. Na BIND 9 sa implementacijom izvršavanje RRL, potrebno je uneti sledeće izmene u konfiguraciji DNS servera:

```
rate-limit {
    responses-per-second 5;
    window 5;
};
```

7.6.2 Microsoft DNS Server

Microsoft DNS server trenutno nema ovu funkcionalnost.

7.7 Konfigurisanje UDP *Source Port Randomization* funkcionalnosti

U nastavku su date konfiguracije BIND-a sa kojima će DNS server nasumično raspoređivati UDP izvorišni port. Date konfiguracije treba primeniti u sekciji "options" "named.conf" konfiguracionog fajla.

```
Configuration UDP Source Port Randomization
// The 'query-source' and 'query-source-v6' configurations
// option allows the operator to select the interface(s)
// and UDP source port value used for sending DNS queries.
// If a value of '*' is used for the source port, then a
// port will be used from pool of random unprivileged ports.
// Query port pools are used by default unless a port value
// is explicitly configured.
//
options {
    query-source address * port *;

    query-source-v6 address * port *;
```

```
// Additional configuration options are available for UDP
// source port randomization. This is achieved through the
// "queryport" options added to version 9.5 of BIND.
//    * use-queryport-pool: Enabled by default unless the
//      port value is explicitly configured for the query-
//      source or query-source-v6 options configuration.
//    * queryport-pool-ports: Defines how many random ports
//      the pool will contain. The default is 8 ports.
//    * queryport-pool-updateinterval: Defines in minutes
//      when the query port pool will be recreated (select
//      a new group of random unprivileged ports). The
//      default is 15 minutes.
//
// By increasing the number of ports allocated to the query
// port pool, it will be harder for malicious users to predict
// the next UDP source port used in DNS queries. Operators
// may also decrease the time interval for the recreation of
// query port pool, thus allowing for random ports to be
// selected in shorter intervals and making predictability
// of source port values harder to determine.
//
// Note: Operators should test any non-default changes prior
// to deploying to production environments.

    queryport-pool-ports <number>;
    queryport-pool-updateinterval <number>;

};
```

7.8 Kontrola veličine i trajanja čuvanja podataka u kešu

Na DNS serveru, kako bi se podesila veličina i vreme trajanja čuvanja podataka u kešu, mogu se koristiti sledeće opcione konfiguracije za BIND. Navedene komande poništavaju vrednost TTL parametra koji se dobija u okviru DNS odgovora, ukoliko je ta vrednost veća od vrednosti zadate ovom komandom.

```
Maximum Cache Length for RRs
// The 'max-cache-ttl' configurations option allows the
// operator to define the amount of time the DNS server
// will store RR information in the resolver cache.
//
// Note: Operators should test any non-default changes
// prior to deploying to production environments.

options {

    max-cache-ttl <number>;

};

Maximum Cache Size
// The 'max-cache-size' configurations option allows the
// operator to define the amount memory a DNS server will
// use for storing RR information in the resolver cache.
// When data stored in cache has reached the configured
// memory limit, BIND will purge RR information from the
// cache to store new RR information.
//
// Note: If this options configuration is set to a low
// value, it may cause the DNS server to issue queries
// more often since entries stored in the cache will be
// purged quicker. This is dependent on the amount of
// queries the DNS server processes.
```

```
//
// Note: Operators should test any non-default changes prior
// to deploying to production environments.

options {

    max-cache-size <number>;

};
```

7.9 Primena *access* listi za kontrolu pristupa DNS serveru

U tabeli T-1 dat je spisak komandi za primenu *access* listi kod BIND softvera u cilju preciznije kontrole prava pristupa u korišćenju DNS servisa. Primenom ovih komandi administrator može da veoma precizno definiše koje IP adrese mogu da izvršavaju određene aktivnosti u pretraživanju DNS prostora.

Tabela T-1 - Spisak *access* list komandi za BIND

Sintaksa komande	Na koje se DNS transakcije odnosi
allow-query { address_match_list }	DNS *Query/Response*
allow-recursion { address_match_list }	*Recursive Query*
allow-transfer { address_match_list }	*Zone Transfer*
allow-update { address_match_list }	*Dynamic Update*
allow-update-forwarding { address_match_list }	*Dynamic Update*
allow-notify { address_match_list }	DNS NOTIFY
blackhole { address_match_list }	*Blacklisted Hosts*

- *allow-query* - definiše listu hostova kojima je dozvoljeno da šalju upite DNS serveru;
- *allow-recursion* - definiše listu hostova kojima je dozvoljeno da pošalju upit sa zahtevom da se on rekurzivno razreši i odgovori na njega;
- *allow-transfer* - definiše listu hostova kojima je dozvoljeno da zahtevaju transfer zonskog fajla;
- *allow-update* - definiše listu hostova kojima je dozvoljeno da šalju DNS Dynamic pakete;
- *allow-update-forwarding* - definiše listu hostova kojima je dozvoljeno da proslede (*forward*) dinamički zahtev;
- *allow-notify* - definiše listu hostova sa kojih je dozvoljeno primiti *NOTIFY* poruku;
- *blackhole* - definiše listu hostova kojima je zabranjena bilo kakva komunikacija sa DNS serverom.

7.10 Aktiviranje DNSSEC validacije odgovora

7.10.1 BIND 9

BIND DNS server podržava DNSSEC već od verzije 9.7 softvera. Današnja verzija BIND softvera (9.10) već stiže sa uključenom podrškom za DNSSEC koja se ogleda u tome da je BIND softver kompajliran sa uključenom podrškom za OpenSSL. To se proverava zadavanjem sledeće komange:

```
$ named -V
BIND 9.10.1 <id:fe66c6b1> built by make with '--prefix=/opt/local'
'--mandir=/opt/local/share/man' '--with-openssl=/opt/local'
'--with-libxml2=/opt/local' '--without-libjson' '--enable-threads'
'--enable-ipv6' 'CC=/usr/bin/clang' 'CFLAGS=-pipe -Os -arch x86_64'
'LDFLAGS=-L/opt/local/lib -Wl,-headerpad_max_install_names -arch x86_64'
'CPPFLAGS=-I/opt/local/include'
compiled by CLANG 4.2.1 Compatible Apple LLVM 5.1 (clang-503.0.40)
using OpenSSL version: OpenSSL 1.0.1i 6 Aug 2014
using libxml2 version: 2.9.1
```

Ukoliko je u softver uključena sva potrebna podrška, aktiviranje verifikacije potpisanih odgovara se vrši sledećom komandom u okviru konfiguracije BIND-a:

```
options {
    dnssec-validation auto;
};
```

Da bi sistem verifikacije digitalnih potpisao mogao da funkcioniše, potreban je prvi čvor u lancu poverenja, nazvan *Trusted Anchor*. Izbor *Trusted Anchor*-a se sastoji u instalaciji njegovog potpisa i sertifikata na serveru tako da DNS server ima početne elemente za DNSSEC validaciju. Izborom opcije *auto* prilikom aktiviranja DNSSEC validacije, posao oko instalacije *Trusted Anchor*-a i njegovo periodično obnavljanje je u potpunosti prepušteno BIND-u.

7.10.2 Microsoft DNS server

Microsoft je tek od Windows Servera 2012 uveo podršku za DNSSEC. Ta podrška je uvedena u potpunosti u skladu sa standardima tako da je njeno korišćenje pravolinijsko. Da bi se na MS Windows Serveru 2012 aktivirala validacija DNSSEC potpisa, potrebno je definisati tzv. *Trust Anchor*. To je prvi čvor u lancu poverenja koji se koristi za verifikaciju digitalnih potpisa. Sa linka [9] mogu da se preuzmu svi potrebni podaci (potpisi i sertifikati) za definisanje *Trust Anchor*-a na lokalnom DNS serveru. Postoje tri načina da se instalira *Trust Anchor* na Windows Serveru 2012: u okviru DNS Manager-a, pomoću Dnscmd.exe programa i u okviru Windows Powershell-a. Ovde je dat primer kako se to radi pomoću Dnscmd.exe programa:

```
PS C:\Users\Administrator> Dnscmd.exe /RetrieveRootTrustAnchors
Are you sure you want to Retrieve and add root trust anchors (activating
DNSSEC validation)? (y/n) y

 The root trust anchors were succesfully retrieved and added as DS trust
anchors.
 They will become effective when the server is able to convert them to
DNSKEY
 trust anchors during an active refresh.

Command completed successfully.

PS C:\Users\Administrator>
```

Po izvršenoj instalaciji *Trusted Anchor*-a, potrebno je aktivirati DNSSEC validaciju na sledeći način:

```
DnsCmd.exe <Servername> /Config /enablednssec 1
```

Ovime je završena aktivacija DNSSEC validacije. Na kraju je potrebno proveriti da li sve funkcioniše kako treba što se najlakše obavlja korišćenjem *dig* programa na sledeći način:

```
dig +dnssec . ns
```

U odgovoru na prethodno navedeni *dig* upit potrebno je da bude setovan *flag ad* što znači *Authenticated Data* što je potvrda da je DNSSEC validacija uspešno urađena.

8 Hardverski zahtevi za korišćenje softvera za DNS server

Zahtevi po pitanju hardvera potrebnog za instalaciju BIND 9.10 softvera u velikoj meri zavise od veličine zonskih fajlova koji će biti hostovani na DNS serveru i od veličine keša za keširanje DNS odgovora. Prema navodima autora BIND-a, za male zone, bez keširanja, može da posluži i računar na bazi Intel i486 procesora.

Kapacitet RAM memorije treba da bude takav da može da prihvati kompletne podatke svih domena koji se hostuju na tom DNS serveru zajedno sa delom memorije za keširanje dobijenih odgovora. Ukoliko nije na raspologanju dovoljno RAM memorije, tada se veličina keša za keširanje odgovora može limitirati primenom komande **max-cache-size**.

Za domene koji koriste DNSSEC potreban je snažan server *enterprise* klase da bi mogao da odradi sve potrebne poslove. Microsoft je testirao svoj softver za DNS servis na Windows Serveru 2013 i objavio rezultate date u tabeli T-2. Pored podataka navedenih u tabeli T-2, poznato je da tipična konfiguracija za *root* DNS server ima Intel Xeon procesor E3-1220 na 3,1GHz sa 16GB RAM memorije i SATA 6Gb/s HDD-ovima vezanim u RAID. Između ova dva primera mogu se smestiti konfiguracije većine ostalih DNS servera, osim DNS servera velikih *Content Distribution Network* operatora koji imaju specifične sisteme za realizaciju DNS servisa.

Tabela T-2 - Performanse rada DNS servera [7]

Konfiguracija servera	Broj upita u sekundi	Broj dinamičkih *update*-a u sekundi	Opterećenje procesora
Intel P-III 733MHz jedan procesor, RAM 256MB, HDD 4GB	9500	1300	75%

Kada se govori o performansama DNS servera, potrebno je razlikovati nekoliko slučajeva:

- autoritativni DNS server,
- rekurzivni DNS server,
- autoritativni DNS server sa DNSSEC podrškom,
- rekurzivni DNS server sa DNSSEC podrškom.

8.1 Autoritativni DNS server

Kod autoritativnig DNS servera, najveće opterećenje u svakodnevnom radu predstavlja odgovaranje na upite. Navedena operacija nije preterano procesorski zahtevna. S druge strane, potrebno je da server ima dovoljno RAM memorije da bi u nju mogao da smesti kompletne podatke iz svih zonskih fajlova za domene za koje je server autoritativni. Pošto ovi serveri ne obrađuju rekurzivne upite, oni nemaju ni memorijski *cache* za smeštanje odgovora. To znači da jedini parametar koji utiče na performanse rada servera je broj upita koji stižu i koje treba obraditi.

8.2 Rekurzivni DNS server

Kod rekurzivnog DNS servera, najveće opterećenje u svakodnevnom radu predstavlja rekurzivno slanje upita dok se ne dobije konačan odgovor koji će biti prosleđen krajnjem korisniku. Pored opterećenja procesora, na ovom tipu DNS servera posebno je važna količina RAM memorije koja je na raspolaganju. RAM memorije se koristi za smeštanje kako podataka iz sopstvenih zonskih fajlova (ako ih ima), tako i za smeštanje odgovora na DNS upite, tj. za keširanje. Što je veća memorija za keširanje, DNS server će moći da čuva više informacija i brže da odgovara na upite.

8.3 Autoritativni DNS server sa DNSSEC podrškom

Uvođenje DNSSEC-a donosi dodatne zahteve po pitanju hardverskih resursa potrebnih za ispravan rad DNS servisa. U slučaju autoritativnih DNS servera to se ogleda u potrebi za digitalnim potpisivanjem svih RR-ova koji se nalaze u zonskim fajlovima. U zavisnosti od učestanosti izmena sadržaja zonskih fajlova, ovaj novi zadatak može da zahteva ozbiljne dodatne hardverske resurse ili može da se reši sa postojećim resursima. Digitalno potpisivanje se obavlja samo kada dođe do izmene RR-ova. Ako su zonski fajlovi veoma veliki, tada je i potpisivanje veoma hardveski zahtevno. Ovaj problem može da se reši korišćenjem eksternog hardvera koji će se koristiti isključivo za digitalno potpisivanje RR-ova ili korišćenjem jačeg hardvera na samom DNS serveru. Pored procesa digitalnog potpisivanja RR-ova, ostatak vremena autoritativni DNS server sa DNSSEC podrškom radi kao i svaki drugi autoritativni DNS server. Jedina razlika je što su DNS odgovori ovog servera nešto duži jer mora da šalje i dodatne RR-ove koji su deo DNSSEC arhitekture.

8.4 Rekurzivni DNS server sa DNSSEC podrškom

Ova vrsta DNS servera ima najveće zahteve po pitanju hardvera. Da bi DNSSEC ispravno funkcionisao, prilikom dobijanja odgovora, ako je on digitalno potpisan, ovaj DNS server mora da proveri digitalni potpis. A to znači da se za svaki takav RR koji prođe kroz njega, pokreće kriptografski modul koji radi proveru. Izvršavanje kriptografskog softvera je, po pravilu, procesorski zahtevna operacija tako da ova vrsta DNS servera ima najveće zahteve po pitanju korišćenog hardvera.

9 Alati za proveru rada DNS servera

Danas na Internetu postoji veliki broj *on-line* alata za proveru rada DNS servera. Međutim, administratori najčešće koriste dva alata koji stižu uz većinu operativnih sistema ili se lako instaliraju. To su *nslookup* i *dig*.

9.1 *nslookup*

Nslookup predstavlja jedan od prvih alata koji je standardno bio deo većine modernih operativnih sistema. Omogućava veoma lako zadavanje DNS upita i proveru odgovora koji se dobijaju. Prilagođen je početnicima koji ne poznaju rad DNS servisa u potpunosti pa je shodno tome i korisnički interfejs krajnje uprošćen. Prilikom ispisa odgovora, on je krajnje uprošćen bez detaljnih informacija po pitanju samog odgovora. Posledica toga je da *nslookup* vrlo često pogrešno protumači odgovor koji je dobijen iz čega korisnik donosi pogrešan zaključak.

Pokretanjem *nslookup*-a u nekom *shell* okruženju, dobija se *nslookup* komandni mod u okviru kojeg korisnik može da zadaje upite. Po *default*-u, ti upiti se šalju lokalnom DNS serveru. Primer običnog DNS upita dat je na slici 4.

```
C:\nslookup
> www.etf.bg.ac.rs
Server:         147.91.8.62
Address:        147.91.8.62#53

www.etf.bg.ac.rs        canonical name = vhost4.etf.bg.ac.rs.
Name:   vhost4.etf.bg.ac.rs
Address: 147.91.14.197
>
```

Slika 4 - *nslookup* - Primer standardnog DNS upita

Pored običnog upita za dobijanje IP adrese, *nslookup* omogućava generisanje još nekih tipova DNS upita koji su uobičajeni za standardni DNS (bez ekstenzija poput DNSSEC). Na slici 5 prikazan je skup parametara koji se mogu zadavati u okviru *nslookup*-a.

```
> help
Commands:   (identifiers are shown in uppercase, [] means optional)
NAME           - print info about the host/domain NAME using default server
NAME1 NAME2     - as above, but use NAME2 as server
help or ?       - print info on common commands
set OPTION      - set an option
   all              - print options, current server and host
   [no]debug          - print debugging information
   [no]d2             - print exhaustive debugging information
   [no]defname         - append domain name to each query
   [no]recurse        - ask for recursive answer to query
   [no]search         - use domain search list
   [no]vc             - always use a virtual circuit
   domain=NAME        - set default domain name to NAME
   srchlist=N1[/N2/.../N6] - set domain to N1 and search list to N1,N2, etc.
   root=NAME          - set root server to NAME
   retry=X            - set number of retries to X
   timeout=X          - set initial time-out interval to X seconds
   type=X             - set query type (ex. A,AAAA,A+AAAA,ANY,CNAME,MX,NS,PTR,
SOA,SRV)
   querytype=X        - same as type
   class=X            - set query class (ex. IN (Internet), ANY)
   [no]msxfr          - use MS fast zone transfer
   ixfrver=X          - current version to use in IXFR transfer request
server NAME     - set default server to NAME, using current default server
lserver NAME    - set default server to NAME, using initial server
root           - set current default server to the root
ls [opt] DOMAIN [> FILE] - list addresses in DOMAIN (optional: output to FILE)
   -a         -  list canonical names and aliases
   -d         -  list all records
   -t TYPE     -  list records of the given RFC record type (ex. A,CNAME,MX,NS,
PTR etc.)
view FILE          - sort an 'ls' output file and view it with pg
exit           - exit the program
```

Slika 5 - Skup parametara koje je moguće podesiti u okviru *nslookup*-a

Da bi se dobio *mail* server za neki domen, sa *nslookup*-om se upit zadaje na način prikazan na slici 6. Da bi se dobilo ime za neku IP adresu, primer *nslookup* upita dat je na slici 7. Da bi se odredio DNS server za neki domen, primer *nslookup* upita dat je na slici 8.

```
> set q=mx
> etf.bg.ac.rs
Server:         147.91.8.62
Address:        147.91.8.62#53

etf.bg.ac.rs    mail exchanger = 5 mail.etf.bg.ac.rs.
> set q=a
> mail.etf.bg.ac.rs
Server:         147.91.8.62
Address:        147.91.8.62#53

Name:   mail.etf.bg.ac.rs
Address: 147.91.14.160
```

Slika 6 - *nslookup* - Određivanje mail servera

```
> set q=ptr
> 147.91.14.160
Server:         147.91.8.62
Address:        147.91.8.62#53

160.14.91.147.in-addr.arpa      name = smtp.etf.bg.ac.rs.
160.14.91.147.in-addr.arpa      name = smtp.etf.rs.
160.14.91.147.in-addr.arpa      name = zmaj.etf.bg.ac.rs.
160.14.91.147.in-addr.arpa      name = zmaj.etf.rs.
160.14.91.147.in-addr.arpa      name = mail.etf.bg.ac.rs.
160.14.91.147.in-addr.arpa      name = mail.etf.rs.
160.14.91.147.in-addr.arpa      name = orao.etf.rs.
```

Slika 7 - *nslookup* - Određivanje imena na osnovu IP adrese

```
> set q=ns
> etf.bg.ac.rs
Server:         147.91.8.62
Address:        147.91.8.62#53

etf.bg.ac.rs    nameserver = ns1.nic.rs.
etf.bg.ac.rs    nameserver = ns2.etf.bg.ac.rs.
etf.bg.ac.rs    nameserver = ns.etf.bg.ac.rs.
etf.bg.ac.rs    nameserver = ns.rcub.bg.ac.rs.
```

Slika 8 - *nslookup* - Određivanje DNS servera za domen

Problem u radu sa *nslookup*-om nastaje kada se postavi neodgovarajući upit. Umesto da pošalje kompletan odgovor iz koga se može zaključiti šta je problem, *nslookup* vraća informaciju da domen ne postoji ili da DNS server uopšte nije odgovorio. Na slici 9 prikazan je jedan takav primer. U primeru na slici 9 prvo je postavljeno da se upiti šalju *root* DNS serveru, a zatim je poslat upit za domen etf.bg.ac.rs. Odgovor *nslookup*-a je bio da *root* DNS server nije uopšte nije odgovorio a u stvarnosti, *root* DNS server je odgovorio da on nije autoritativni za navedeni domen i da ne izvršava rekurzivne DNS upite. Ništa od navedenog

se ne vidi u odgovoru *nslookup*-a. Pored navedenog problema, *nslookup* ne može da se koristi za proveru DNSSEC funkcionalnosti.

```
> server i.root-servers.net
Default server: i.root-servers.net
Address: 192.36.148.17#53
Default server: i.root-servers.net
Address: 2001:7fe::53#53
> etf.bg.ac.rs
Server:         i.root-servers.net
Address:        192.36.148.17#53

Non-authoritative answer:
*** Can't find etf.bg.ac.rs: No answer
```

Slika 9 - *nslookup* - Odgovor na neodgovarajući upit

9.2 *dig*

Dig je alat koji je novijeg datuma i stiže uz instalaciju BIND softverskog paketa. Pored toga, sve češće Linux instalacije sadrže i *dig*. Osnovna karakteristika *dig*-a je da je namenjen korišćenju od strane iskusnih korisnika koji dobro poznaju DNS servis i način njegovog rada. Omogućava detaljnu proveru rada DNS servisa, kako u standardnom režimu, tako i kada se koristi DNSSEC. Na slikama 10 do 14 ponovljeni su primeri koji su prezentirani za *nslookup*.

```
$ dig @ns.etf.rs www.etf.bg.ac.rs a

; <<>> DiG 9.X.X-RedHat<<>> @ns.etf.rs www.etf.bg.ac.rs a
; (1 server found)
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 61456
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 4, ADDITIONAL: 5

;; QUESTION SECTION:
;www.etf.bg.ac.rs.            IN      A

;; ANSWER SECTION:
www.etf.bg.ac.rs.       3600    IN      CNAME   vhost4.etf.bg.ac.rs.
vhost4.etf.bg.ac.rs.    3600    IN      A       147.91.14.197

;; AUTHORITY SECTION:
etf.bg.ac.rs.           3600    IN      NS      ns.rcub.bg.ac.rs.
etf.bg.ac.rs.           3600    IN      NS      ns2.etf.bg.ac.rs.
etf.bg.ac.rs.           3600    IN      NS      ns.etf.bg.ac.rs.
etf.bg.ac.rs.           3600    IN      NS      ns1.nic.rs.

;; ADDITIONAL SECTION:
ns1.nic.rs.             86400   IN      A       147.91.8.6
ns.etf.bg.ac.rs.        3600    IN      A       147.91.8.6
ns2.etf.bg.ac.rs.       3600    IN      A       147.91.8.62
ns.rcub.bg.ac.rs.       3600    IN      A       147.91.1.5
ns.rcub.bg.ac.rs.       3600    IN      AAAA    2001:4170:0:1::5

;; Query time: 2 msec
;; SERVER: 147.91.8.6#53(147.91.8.6)
;; WHEN: Wed Dec  2 16:46:00 2015
;; MSG SIZE  rcvd: 242
```

Slika 10 - *dig* - Primer standardnog DNS upita

```
$ dig @ns.etf.rs etf.bg.ac.rs mx

; <<>> DiG 9.X.X-RedHat<<>> @ns.etf.rs etf.bg.ac.rs mx
; (1 server found)
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 64459
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 4, ADDITIONAL: 6

;; QUESTION SECTION:
;etf.bg.ac.rs.                IN      MX

;; ANSWER SECTION:
etf.bg.ac.rs.          3600    IN      MX      5 mail.etf.bg.ac.rs.

;; AUTHORITY SECTION:
etf.bg.ac.rs.          3600    IN      NS      ns.rcub.bg.ac.rs.
etf.bg.ac.rs.          3600    IN      NS      ns2.etf.bg.ac.rs.
etf.bg.ac.rs.          3600    IN      NS      ns.etf.bg.ac.rs.
etf.bg.ac.rs.          3600    IN      NS      ns1.nic.rs.

;; ADDITIONAL SECTION:
mail.etf.bg.ac.rs.     3600    IN      A       147.91.14.160
ns1.nic.rs.            86400   IN      A       147.91.8.6
ns.etf.bg.ac.rs.       3600    IN      A       147.91.8.6
ns2.etf.bg.ac.rs.      3600    IN      A       147.91.8.62
ns.rcub.bg.ac.rs.      3600    IN      A       147.91.1.5
ns.rcub.bg.ac.rs.      3600    IN      AAAA    2001:4170:0:1::5

;; Query time: 1 msec
;; SERVER: 147.91.8.6#53(147.91.8.6)
;; WHEN: Wed Dec  2 17:16:13 2015
;; MSG SIZE  rcvd: 238
```

Slika 11 - *dig*- Određivanje *mail* servera

```
$ dig @ns.etf.rs 160.14.91.147.in-addr.arpa ptr

; <<>> DiG 9.X.X-RedHat<<>> @ns.etf.rs 160.14.91.147.in-addr.arpa ptr
; (1 server found)
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 35249
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 7, AUTHORITY: 3, ADDITIONAL: 4

;; QUESTION SECTION:
;160.14.91.147.in-addr.arpa.    IN      PTR

;; ANSWER SECTION:
160.14.91.147.in-addr.arpa. 86400 IN    PTR     orao.etf.rs.
160.14.91.147.in-addr.arpa. 86400 IN    PTR     zmaj.etf.rs.
160.14.91.147.in-addr.arpa. 86400 IN    PTR     zmaj.etf.bg.ac.rs.
160.14.91.147.in-addr.arpa. 86400 IN    PTR     mail.etf.rs.
160.14.91.147.in-addr.arpa. 86400 IN    PTR     mail.etf.bg.ac.rs.
160.14.91.147.in-addr.arpa. 86400 IN    PTR     smtp.etf.rs.
160.14.91.147.in-addr.arpa. 86400 IN    PTR     smtp.etf.bg.ac.rs.

;; AUTHORITY SECTION:
14.91.147.in-addr.arpa. 86400   IN      NS      ns2.etf.rs.
14.91.147.in-addr.arpa. 86400   IN      NS      NS.etf.rs.
14.91.147.in-addr.arpa. 86400   IN      NS      NS.RCUB.bg.ac.rs.

;; ADDITIONAL SECTION:
NS.etf.rs.          3600    IN      A       147.91.8.6
ns2.etf.rs.         3600    IN      A       147.91.8.62
NS.RCUB.bg.ac.rs.       3600    IN      A       147.91.1.5
NS.RCUB.bg.ac.rs.       3600    IN      AAAA    2001:4170:0:1::5

;; Query time: 1 msec
;; SERVER: 147.91.8.6#53(147.91.8.6)
;; WHEN: Wed Dec  2 17:18:37 2015
;; MSG SIZE  rcvd: 326
```

Slika 12 - *dig* - Određivanje imena na osnovu IP adrese

```
$ dig @ns.etf.rs etf.rs ns

; <<>> DiG 9.X.X-RedHat<<>> @ns.etf.rs etf.rs ns
; (1 server found)
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 34098
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 4, AUTHORITY: 0, ADDITIONAL: 5

;; QUESTION SECTION:
;etf.rs.                        IN      NS

;; ANSWER SECTION:
etf.rs.             3600    IN      NS      ns.rcub.bg.ac.rs.
etf.rs.             3600    IN      NS      ns2.etf.rs.
etf.rs.             3600    IN      NS      ns1.nic.rs.
etf.rs.             3600    IN      NS      ns.etf.rs.

;; ADDITIONAL SECTION:
ns1.nic.rs.          86400   IN      A       147.91.8.6
ns.etf.rs.           3600    IN      A       147.91.8.6
ns2.etf.rs.          3600    IN      A       147.91.8.62
ns.rcub.bg.ac.rs.    3600    IN      A       147.91.1.5
ns.rcub.bg.ac.rs.    3600    IN      AAAA    2001:4170:0:1::5

;; Query time: 0 msec
;; SERVER: 147.91.8.6#53(147.91.8.6)
;; WHEN: Wed Dec  2 17:20:45 2015
;; MSG SIZE  rcvd: 201
```

Slika 13 - *dig* - Određivanje DNS servera za domen

```
$ dig @i.root-servers.net etf.bg.ac.rs

; <<>> DiG 9.X.X-RedHat <<>> @i.root-servers.net etf.bg.ac.rs
; (2 servers found)
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 62531
;; flags: qr rd; QUERY: 1, ANSWER: 0, AUTHORITY: 8, ADDITIONAL: 14

;; QUESTION SECTION:
;etf.bg.ac.rs.                  IN      A

;; AUTHORITY SECTION:
rs.                     172800  IN      NS      k.nic.rs.
rs.                     172800  IN      NS      b.nic.rs.
rs.                     172800  IN      NS      d.nic.rs.
rs.                     172800  IN      NS      h.nic.rs.
rs.                     172800  IN      NS      f.nic.rs.
rs.                     172800  IN      NS      l.nic.rs.
rs.                     172800  IN      NS      a.nic.rs.
rs.                     172800  IN      NS      g.nic.rs.

;; ADDITIONAL SECTION:
a.nic.rs.               172800  IN      A       91.199.17.59
a.nic.rs.               172800  IN      AAAA    2001:67c:69c::59
b.nic.rs.               172800  IN      A       195.178.32.2
d.nic.rs.               172800  IN      A       193.0.9.107
d.nic.rs.               172800  IN      AAAA    2001:67c:e0::107
f.nic.rs.               172800  IN      A       204.61.216.32
f.nic.rs.               172800  IN      AAAA    2001:500:14:6032:ad::1
g.nic.rs.               172800  IN      A       147.91.8.6
h.nic.rs.               172800  IN      A       91.199.17.60
h.nic.rs.               172800  IN      AAAA    2001:67c:69c::60
k.nic.rs.               172800  IN      A       192.5.4.1
k.nic.rs.               172800  IN      AAAA    2001:500:2e::1
l.nic.rs.               172800  IN      A       194.146.106.114
l.nic.rs.               172800  IN      AAAA    2001:67c:1010:29::53

;; Query time: 1 msec
;; SERVER: 192.36.148.17#53(192.36.148.17)
;; WHEN: Wed Dec  2 17:22:15 2015
;; MSG SIZE  rcvd: 458
```

Slika 14 - *dig* - Odgovor na neodgovarajući upit

Na primeru sa slike 14 vidimo da je *root* DNS server ipak nešto odgovorio, za razliku od *nslookup*-a (primer sa slike 9). Konkretno, *root* DNS server je vratio informaciju o DNS serverima za .rs *top-level* domen pošto je domen etf.bg.ac.rs deo .rs domena. Pored IP adresa i DNS servera koji su autoritativni za davanje odgovora na pitanje, *dig* prikazuje i *flag*-ove koji su se nalazi u DNS paketu. Ti *flag*-ovi su prikazani sa *qr, rd¸aa, tc, ra*. Osnovno tumačenje ovih *flag*-ova dato je u RFC-u 1035, ovde je dat samo skraćeni prikaz:

- *qr - Query Response* - označava da se radi o odgovoru na upit,
- *aa - Authoritative Answer* - označava da se radi o odgovoru dobijenom od autoritativnog servera,
- *tc - TrunCated* - označava da nije prikazan ceo odgovor jer nije mogao da stane u UDP paket i da treba koristiti TCP protokol za dobijanje odgovora od DNS servera,
- *rd - Recursion Desired* - označava da je klijent zahtevao od servera da uradi rekurzivno određivanje traženog RR-a,
- *ra - Recursion Available* - označava da je odgovor dobijen od rekurzivnog DNS servera.

Pored prethodno navedenih tumačenja, *dig* ispisuje podatke i o statusu odgovora koji je dobijen (detaljno objašnjenje postoji u RFC-u 1035), a ovde je dat samo skraćeno prikaz:

- NOERROR - nema grešaka u komunikaciji,
- FORMERR - greška u formatu upita,
- SRVFAIL - *SeRVer FAILure* - server nije mogao da odgovori na upit zbog raznih grešaka do kojih je došlo,
- NXDOMAIN - domen za koji je poslat upit ne postoji na ovom DNS serveru,
- NOTIMPL - *NOT IMPLemented* - server ne razume tip upita koji je poslat,
- REFUSED - server je odbio da odgovori na upit.

10 Literatura

[1.] ISC Domain Survey, https://www.isc.org/services/survey/
[2.] *DNS Best Practices, Network Protections and Attack Identification* http://www.cisco.com/web/about/security/intelligence/dns-bcp.html
[3.] http://resources.infosecinstitute.com/dns-tunnelling/
[4.] https://www.us-cert.gov/ncas/alerts/TA13-088A
[5.] Ramaswamy Chandramouli, Scott Rose: *Secure Domain Name System (DNS) Deployment Guide*, NIST Special Publication 800-81
[6.] SAC065 - *SAC Advisory on DDoS Attacks Leveraging DNS Infrastructure*, ICANN Security and Stability Advisory Committee, 18.02.2014.
[7.] *Monitoring DNS server performance,* https://technet.microsoft.com/en-us/library/cc778608%28v=ws.10%29.aspx
[8.] RIPE 352: *Measuring The Resource Requirements Of DNSSEC*, https://www.ripe.net/publications/docs/ripe-352
[9.] IANA *root zone Trust Anchor,* http://data.iana.org/root-anchors/
[10.] Spisak RFC preporuka relevantnih za DNS servis, https://www.isc.org/community/rfcs/dns/