ZAVOD ZA ELEKTRONIKU, MIKROELEKTRONIKU, RAČUNALNE I INTELIGENTNE SUSTAVE
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA
SVEUČILIŠTE U ZAGREBU

DIPLOMSKI RAD br. 1784

Analiza i prikupljanje DNS paketa

Dinko Korunić

Zagreb, veljača 2009.

Sažetak

Predmet promatranja ovog diplomskog rada je područje DNS protokola vezano uz brojne kritične sigurnosne prijetnje prema DNS poslužiteljima. U radu se razmatra izrada sustava distribuiranog pasivnog prisluškivanja DNS komunikacije uz istovremenu opću i sigurnosnu analizu navedenog prometa, identifikaciju sigurnosnih problema te predstavljanje rezultata korisniku. Uz razradu DNS problematike i pojedinosti sustava za analizu DNS prometa te formalno testiranje sukladnosti standardima, obavljena su i praktična mjerenja na centralnim DNS poslužiteljima Zavoda za elektroniku, mikroelektroniku, računalne i inteligentne sustave Fakulteta elektrotehnike i računarstva u Zagrebu te Fakulteta strojarstva i brodogradnje u Zagrebu.

Ključne riječi

DNS protokol, DNS trovanje, analiza DNS prometa, sustavi za otkrivanje neovlaštenog upada.

Abstract

This work deals with numerous security threats to the DNS protocol. We are discussing the idea behind the distributed DNS monitoring system which passively monitors the DNS traffic, performs the basic and the security analysis, performs the identification of security issues and presents the results to the end user. The related details of DNS protocols and standards are documented, as well as all necessary prerequisites and components of DNS monitoring system itself. We have performed the formal standards compliance testing and practical DNS data analysis on central DNS servers at Department of Electronics, Microelectronics, Computer and Intelligent Systems of Faculty of Electrical Engineering and Computing and Faculty of Mechanical Engineering and Naval Architecture, Zagreb.

Keywords

DNS protocol, DNS poisoning, DNS traffic analysis, Intrusion Detection Systems.

Sadržaj

1. Uvod....1

2. Imenički sustav domena....3
2.1. Tipovi DNS upita....3
2.2. DNS Resource Record....6
2.3. Tipovi DNS zapisa....7
2.4. DNS upiti i odgovori....9
2.5. Tipovi DNS poslužitelja....12
2.6. Sigurnosni problemi....14
2.7. Metode analize prometa u poslužiteljima....15
2.8. Pregled postojećih specijaliziranih alata....18

3. Sustav za nadzor i analizu DNS prometa....20
3.1. Razrada implementacije....21
3.1.1. Efikasna komunikacija....22
3.1.2. Minimalno opterećenje računala senzora....22
3.1.3. Minimalno opterećenje centralnog poslužitelja....22
3.1.4. Kriptiranje prometa i provjera autentičnosti....23
3.1.5. Autentikacija i autorizacija....25
3.1.6. Prijenos programskih struktura....26
3.2. Komponente i karakteristike sustava....26
3.3. Otkrivanje problematičnog prometa....30
3.4. Daljnji rad....32

4. Rezultati i razmatranje....33
4.1. Formalno testiranje sustava....33
4.2. Mjerenja u produkciji i diskusija rezultata....35

5. Zaključak....47

6. Literatura....49

7. Dodatak A: Sadržaj priloženog medija (CD/DVD)....51

8. Dodatak B: Upute za instalaciju....52

9. Dodatak C: Upute za korištenje....53

Popis oznaka i kratica

DNS	Domain Name System
IP	Internet Protocol
HTTP	Hypertext Transfer Protocol
SMTP	Simple Mail Transfer Protocol
NNTP	Network News Transfer Protocol
LDAP	Lightweight Directory Access Protocol
TLD	Top-level domain
IANA	Internet Assigned Numbers Authority
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
RR	Resource Record
ISP	Internet service provider
FQDN	Fully qualified domain name
TTL	Time to live
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
SPF	Sender Policy Framework
GPS	Global Positioning System
DNSSEC	DNS Security Extensions
ISDN	Integrated Services Digital Network
PSDN	Public switched data network
RFC	Request for Comments
AS	Autonomous system
IPsec	Internet Protocol Security
IDS	Intrusion detection system
ID	Identification
PCAP	Packet Capture
SLD	Second-level domain
3LD	Third-level domain
OSI	Open Systems Interconnection
FIFO	First In, First Out
COW	Copy-on-write
IKE	Internet Key Exchange
PSK	Pre-shared key
AES	Advanced Encryption Standard
CBC	Cipher Block Chaining
ECB	Electronic CodeBook
CFB	Cipher FeedBack
OFB	Output FeedBack
CTR	Counter
HMAC	keyed-Hash Message Authentication Code
MD5	Message-Digest Algorithm 5
SHA1	Secure Hash Algorithm 1
IV	Initialization vector
ICV	Integrity check value
QPS	Queries per second
WPAD	Web Proxy Autodiscovery Protocol

Popis tablica

Tablica 2.1: Odjeljci u DNS paketu....9
Tablica 2.2: Prikaz zaglavlja u DNS paketu....10
Tablica 2.3: Polja u odjeljku upita....12
Tablica 2.4: Pregled analize prometa u DNS poslužiteljima....16
Tablica 2.5: Pregled alata za DNS analizu....18
Tablica 4.1: Referentna konfiguracija Bind poslužitelja....33
Tablica 4.2: Utjecaj nadzora na DNS performanse....35
Tablica 7.1: Sadržaj priloženog medija....51

Popis slika

Slika 2.1: Rezolucija u DNS arhitekturi....4
Slika 2.2: DNS topologija....5
Slika 2.3: Prikaz DNS paketa s odjeljcima....9
Slika 3.1: Pregled komunikacijskog paketa....25
Slika 3.2: Arhitekturalni prikaz komunikacije u sustavu....28
Slika 3.3: Tijek akcija obrade DNS paketa....29
Slika 3.4: Međuodnos programskih klasa....30
Slika 4.1: Raspodjela ukupnog broja incidenata na FSB-u....37
Slika 4.2: RFC1918 upiti (privatne adrese)....38
Slika 4.3: Odgovori na nepostojeće upite....38
Slika 4.4: Odgovori različiti od upita....39
Slika 4.5: Višestruki odgovori na upit....39
Slika 4.6: A-za-A sigurnosni incidenti....40
Slika 4.7: Nedozvoljeni znakovi u upitu....40
Slika 4.8: Nepoznati tip upita....41
Slika 4.9: Povratna pogreška od DNS poslužitelja....41
Slika 4.10: Nepoznate vršne domene....42
Slika 4.11: Skupni prikaz zabilježenih incidenata na FSB-u....43
Slika 4.12: Raspodjela ukupnog broja incidenata na ZEMRIS-u....45
Slika 4.13: Skupni prikaz zabilježenih incidenata na ZEMRIS-u....46

1. Uvod

DNS (eng. *Domain Name System*) je imenički servis koji omogućava povezivanje slovnih naziva na Internetu s IP (eng. *Internet Protocol*) adresama koje jedinstveno adresiraju udaljeni resurs i omogućavaju komunikaciju s njime [1]. DNS je danas jedan od ključnih Internet servisa koji se koristi u velikoj većini ostalih aplikativnih protokola na Internetu. Razlog leži prvenstveno u jednostavnosti korištenja slovnih i lako pamtljivih oznaka umjesto konkretnih IP adresa.

DNS je realiziran kao strogo hijerarhijski distribuirani sustav u kojem se mogu nalaziti različite informacije, ali se prvenstveno koriste one o IP adresama i slovnim nazivima. U DNS imenicima se najčešće nalaze slovni nazivi računala ili uređaja (eng. *hostname*), a svaki takav naziv je jedinstveno simboličko ime unutar pojedine mreže koje služi za elektroničku identifikaciju nekog računala. Takvim se imenom koriste različiti aplikativni protokoli poput HTTP (eng. *Hypertext Transfer Protocol*), SMTP (eng. *Simple Mail Transfer Protocol*), NNTP (eng. *Network News Transfer Protocol*) i sl. Takvi slovni nazivi mogu biti samo jedna riječ, ako se radi o lokalnoj mreži, ili pak nekoliko riječi odvojenih točkama. U potonjem slučaju riječ je o domenskom imenu (eng. *domain name*) koje predstavlja simboličko ime računala zajedno s hijerarhijski poredanim imenima nadređenih (u logičkom, ali ne nužno fizičkom smislu blizine) grupa računala.

DNS poslužitelji pružaju DNS informacije koristeći DNS protokol za komunikaciju kako s klijentima tako i međusobno. U imenike je moguće spremati i razne dodatne informacije poput onih za aplikativno usmjeravanje mrežnih komunikacija, hardverske opise računala, itd. Cjelokupan DNS sustav je puno širi, te obuhvaća tri osnovne funkcije [2]:

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

Predmet promatranja ovog diplomskog rada je područje DNS protokola vezano uz brojne kritične sigurnosne prijetnje prema DNS poslužiteljima. U radu se razmatra izrada sustava distribuiranog pasivnog prisluškivanja DNS komunikacije uz istovremenu opću i sigurnosnu analizu navedenog prometa, identifikaciju sigurnosnih problema te predstavljanje rezultata korisniku.

Poglavlje 2 ima funkciju uvoda u DNS sustav i komunikaciju što je potrebno za razumijevanje problematike i razloga za analizu DNS prometa. U tom se poglavlju razmatraju i nedostatci postojećih alata, te brojni sigurnosni problemi vezani uz DNS. Poglavlje 3 opisuje potrebne karakteristike sustava, dok se poglavlje 4 bavi testiranjem kao i dobivenim rezultatima mjerenja.

2. Imenički sustav domena

Svaki DNS[1] poslužitelj za komunikaciju standardno koristi port 53 koji mu je dodijeljen od IANA-e (eng. *Internet Assigned Numbers Authority*). Na navedenom portu osluškuje RFC793 TCP (eng. *Transmission Control Protocol*) odnosno RFC768 UDP (eng. *User Datagram Protocol*) upite, dok odgovor može poslati bilo s tog istog porta ili nekog drugog visokog porta (port veći od 1024) ovisno o konfiguraciji poslužitelja. Odgovor sadrži status o uspjehu odnosno pogrešci kao i eventualne tražene zapise odnosno RR-ove (eng. *Resource Record*) [3]. U slučaju da poslužitelj nema tražene podatke, ali ima podatke kamo klijent treba dalje nastaviti s upitom, poslat će te dodatne informacije umjesto traženih.

Standardno se koristi UDP za upite, a komunikacija se uglavnom svodi na jedan UDP upit i jedan UDP odgovor. TCP komunikacije se koristi uglavnom kad veličina odgovora prelazi 512 bajtova ili za grupne prijenose DNS informacija, tzv. prijenos zone (eng. *Zone Transfer*). Moguće su i situacije gdje DNS poslužitelj uopće ne odgovara na TCP upite ili čak situacije gdje DNS klijent šalje samo TCP upite.

Pojavom RFC2671 EDNS0 standarda je problem ograničenja UDP DNS paketa na 512 bajtova riješen; klijent i poslužitelj dogovaraju oko veličine prije samog slanja "velikog" paketa korištenjem specijalnog EDNS0 OPT RR upita [4]. Sam standard je unazadno kompatibilan s postojećim implementacijama, iako njegovo uvođenje uvjetuje nadogradnju DNS poslužitelja i DNS klijenata što nerijetko povlači i nadogradnju cjelokupnog operacijskog sustava.

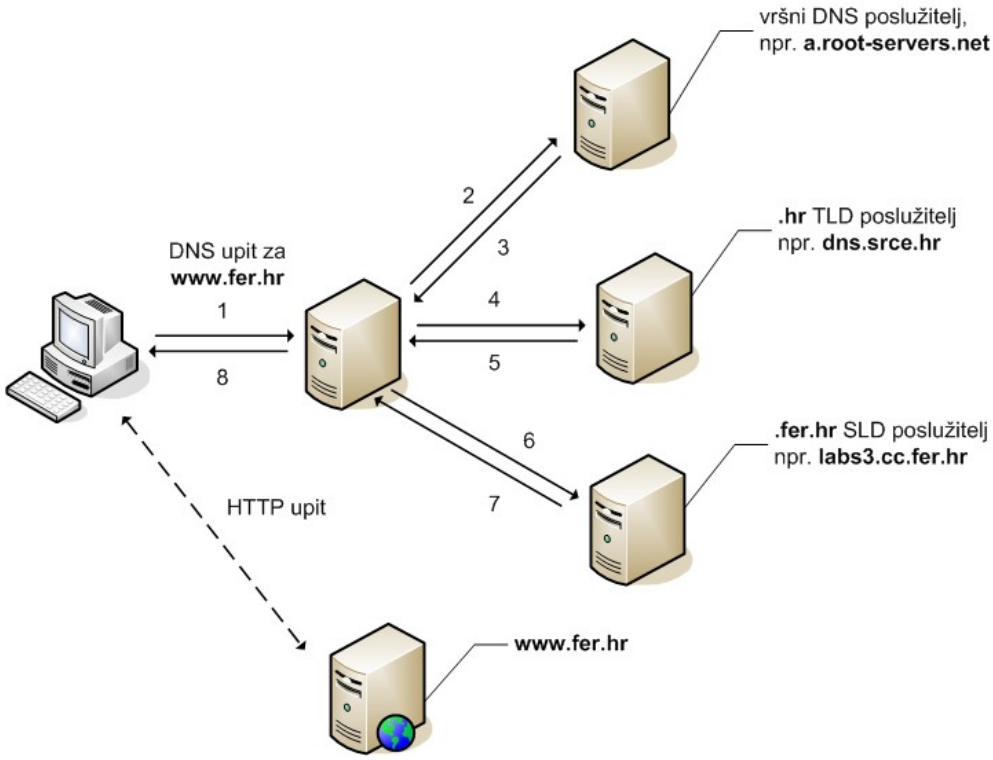
2.1. Tipovi DNS upita

Svaki se funkcionalni DNS sustav nužno sastoji se od tri dijela [5]:

- DNS klijent (eng. *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 (eng. *Recursive*) DNS poslužitelj, koji nakon dobivenih upita za klijenta obavlja pretraživanje kroz DNS stablo i vraća nazad odgovore klijentima,
- Autoritativni (eng. *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.

1 Osnovni DNS standardi za razumijevanje problematike su barem: RFC1034, RFC1035, RFC1101, RFC1123, RFC1183, RFC1591, RFC2181.

Sam proces primanja zahtjeva i njihove obrade te vraćanja odgovora se naziva DNS rezolucija (eng. *Name Resolution*). To je proces pretvorbe domenskog imena u IP adresu: prvo tražimo autoritativni DNS poslužitelj, zatim mu šaljemo upit za adresom na koji on odgovara s traženom adresom. Budući da je DNS strogo distribuirani imenički servis, on je razdijeljen po mnogo različitih poslužitelja. Zbog te 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 rekurzivnom DNS poslužitelju koji je nadležan za mrežu u kojoj se nalazi to klijentsko računalo i koji obavlja zadane upite te zatim vraća odgovor klijentu. Takav poslužitelj je obično dodijeljen od ISP-a (eng. *Internet service provider*) ili ustanove u kojoj se nalazi klijentsko računalo. Najveći i najsloženiji dio rezolucije predstavlja traženje autoritativnog poslužitelja u složenoj DNS hijerarhiji, kao što se može vidjeti na slici 2.1.



Slika 2.1: Rezolucija u DNS arhitekturi

Postoje dva osnovna tipa DNS rezolucije odnosno prolaska kroz DNS hijerarhiju da bi se doznao konkretan 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 odnosno kad lokalni DNS poslužitelj nema sve tražene informacije:

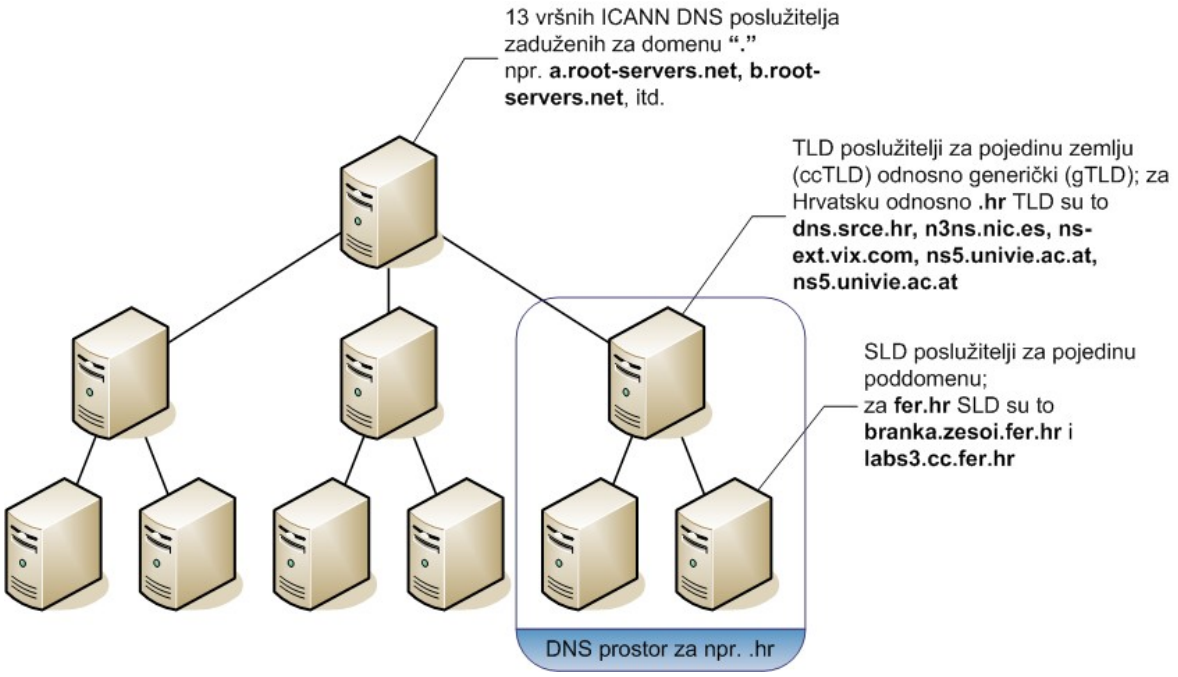
- Iterativni - kada klijent šalje dotične upite, poslužitelj mora odgovoriti jednim od dva moguća odgovora: a) odgovorom na zahtjev ili b) imenom drugog DNS

poslužitelja (obavlja 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. 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. To znači da klijent šalje svega jedan zahtjev te dobiva ili točnu informaciju koju je tražio ili poruku o pogrešci.

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, računalima kojima je dotični DNS poslužitelj nadležan.

DNS stablo je hijerarhijski složen skup DNS poslužitelja, gdje svaka domena i poddomena ima jednog ili više autoritativnih DNS poslužitelja. Dotični poslužitelji (čvorovi stabla) su nadležni za sve domene ispod njih, odnosno odgovaraju na upite direktno sa traženom informacijom ili obavljaju delegiranje prema nekom drugom poslužitelju. Hijerarhijski raspored poslužitelja upravo mora odgovarati rasporedu domena i odgovarajućeg domenskog prostora, kao što je prikazano u slici 2.2.



Slika 2.2: DNS topologija

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 bilo kojeg čvora stabla je lokalna lista od 13 vršnih DNS poslužitelja i njihovih IP adresa. Naime upravo su ti vršni poslužitelji koji dalje delegiraju pretragu po zapisima i bez kojih globalni DNS sustav ne može funkcionirati. Dio adresa vršnih poslužitelja se distribuira anycast tehnologijom kako bi se omogućila decentralizacija i smanjenje opterećenja na pojedinim poslužiteljima. Na taj način se veliki broj distribuiranih čvorova u svijetu pojavljuje kao jedinstveni čvor odnosno servis pri čemu DNS klijenti automatski odabiru najbliži. Trenutno je za obavljanje funkcije 13 vršnih poslužitelja raspoređeno ukupno stotinjak (u veljači 2008. bilo je 169 vršnih poslužitelja prema URL:http://www.root-servers.org/) fizičkih poslužitelja diljem svijeta. Trenutni raspored vršnih DNS poslužitelja je daleko od ravnomjerne distribucije [6]: 38 poslužitelja je nadležno za Sjevernu Ameriku, 35 za Europu, 2 za Australiju, 2 za Novi Zeland, 2 za Kinu, 2 za Rusiju, dok ostale zemlje nemaju vršne DNS poslužitelje, a ponekad ni TLD poslužitelje u svojoj zemlji.

2.2. DNS Resource Record

RR je osnovni zapis odnosno jedinica u DNS sustavu. RR sadrži određene atribute, odgovarajuće za vlastiti tip, a to mogu biti: IP adresa, adresa za isporuku elektroničke pošte, niz znakova, DNS oznaka ili nešto treće. RR se sastoji od sljedećih komponenti, navedenih redom kojim se pojavljuju [7]:

- Ime domene - uglavnom se koristi FQDN (eng. *Fully qualified domain name*), a ako je zapisano kratko ime onda se automatski dodaje ime zone na kraj imena,
- TTL (eng. *Time to live*) 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).

Budući da je od početka bilo zamišljeno da će se kroz DNS nuditi imeničke usluge za više od jednog protokola (dakle i druge protokole osim IP-a), DNS je oformljen vrlo općenito. Stoga svaki RR unutar zone ima i svoju klasu (eng. *Resource Record Classes*), iako su one u osnovi povijesna ostavština. Danas se u praksi koristi jedino Internet klasa, pa se ona implicitno podrazumijeva kad u lokalnoj zoni nije eksplicitno navedena IN klasa.

2.3. Tipovi DNS zapisa

Postoji dosta različitih tipova DNS zapisa koji se prvenstveno razlikuju po svojoj namjeni:

- A (eng. *Address*) - povezuje odgovarajuće domensko ime (oznaku ili niz oznaka) s 32-bitnom IPv4 (eng. *Internet Protocol version 4*) adresom. Danas je često moguće naći da više A zapisa pokazuje na istu IP adresu,
- CNAME (eng. *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 neispravno je u zoni imati ijedan zapis koji dijeli isto ime (oznaku) kao i CNAME zapis; CNAME ne može istodobno postojati niti s jednim drugim tipom za istu oznaku, uključujući i praznu oznaku. 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 (eng. *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 postojanja više e-mail poslužitelja za jednu domenu s točnim redoslijedom prema kojem ih se mora kontaktirati. Time se na jednostavan način omogućava usmjerivanje e-maila (eng. *Mail Routing*) kao i mogućnost raspodjele opterećenja između više poslužitelja. MX zapis ne omogućava postavljanje e-mail servisa na alternativnim portovima niti ne postavljanje težinskih vrijednosti za poslužitelje koji su istog prioriteta kao što SRV zapis omogućava. MX zapis funkcionira tako da klijent pri MX zahtjevu dobiva listu e-mail poslužitelja, te on započinje isporuku pošte na način da je MX zapis s najmanjim pripadnim brojem (eng. *Preference*) onaj s najvećim prioritetom. Klijent tako prolazi listu poslužitelja sve dok uspješno ne isporuči e-mail. Svi poslužitelji koji imaju isti MX broj se tretiraju s jednakim prioritetom, pa se stoga nad njima svima pokušava isporuka dok ne uspije,
- PTR (eng. *Pointer Record*) - povezuje IPv4 adresu s odgovarajućim domenskim imenom odnosno FQDN. Obično PTR zapisi trebaju pokazivati na ime koje se može unazadno razriješiti u polaznu IPv4 adresu. PTR zapis nije IPv4 adresa, već obrnuto zapisana 4 okteta adrese s dodatnom IN-ADDR.ARPA. domenom,
- NS (eng. *Name Server Record*) - označava da za dotičnu zonu treba posluživati upravo dotični DNS poslužitelj. Svaki NS zapis je ili oznaka autoriteta ili oznaka za delegaciju: ako je naziv NS zapisa jednak zoni u kojoj se NS zapis pojavljuje, riječ je o autoritativnom zapisu; ako je pak riječ o nazivu koji sadrži neku od poddomena, riječ je o delegaciji,

- SOA (eng. *Start of Authority*) - označava koji je DNS poslužitelj autoritativan za dotičnu domenu, a donosi i dodatne informacije o zoni. Svaka ispravna zona mora imati SOA zapis,
- AAAA i A6 - povezuju odgovarajuće domensko ime s 128-bitnom IPv6 (eng. *Internet Protocol version 6*) adresom. Moguće je naći i AAAA i A6 zapis, pri čemu se oni razlikuju u nekim detaljima: A6 omogućava da oznaka bude definirana kao binarni niz, itd. Danas se A6 još uvijek smatra eksperimentalnim zapisom, te se u produkciji preporuča koristiti AAAA,
- DNAME (eng. *Delegation Name*) - 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. Ne koristi se u praksi,
- SRV (eng. *Server Selection*) - zapis koji se sve češće koristi u protokolima koji se tek pojavljuju na tržištu, a predstavlja značajno bolju alternativu MX zapisima. Riječ je o općenitom zapisu za definiciju lokacije servisa, njegove težine i prioriteta, primjerice za LDAP (eng. *Lightweight Directory Access Protocol*), HTTP, SMTP i sl,
- TXT (eng. *Text String*) - omogućava proizvoljni tekstualni zapis do 255 bajtova. Danas se koristi primjerice umjesto zastarjelog HINFO opisa uređaja koji nosi domensko ime ili za upisivanje SPF (eng. *Sender Policy Framework*)[2] obilježja,
- DS (eng. *Delegation Signer*) - dodaje se na mjestu prekida zone (mjesta gdje se obavlja 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 standardnog implicitnog načina,
- KEY (eng. *Public Key*) - javni ključ koji je autoriziran od SIG zapisa, a omogućava pohranu i DNSSEC[3] (eng. *DNS Security Extensions*) ključeva i proizvoljnih ključeva za aplikacije,
- KX (eng. *Key Exchanger*) - omogućava metodu za delegiranje autorizacije za neki čvor u ime jednog ili više čvorova, kako bi pružili servise razmjene ključeva,
- LOC (eng. *Location Information*) - zapis u koji je moguće spremiti geolokacijske odnosno GPS (eng. *Global Positioning System*) podatke o određenom čvoru ili domeni,
- SIG (eng. *Cryptographic Public Key Signature*) - predstavlja potpis radi autentifikacije podataka u DNSSEC-u,

2 SPF mehanizam je detaljno dokumentiran u RFC4408, iako je još uvijek riječ o eksperimentalnom protokolu koji nije općeprihvaćeni standard.

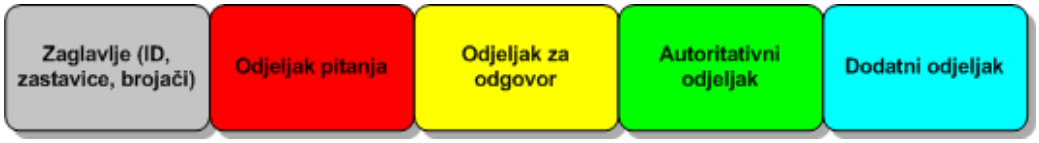
3 DNSSEC je cijeli set ekstenzija na osnovni DNS standard (više o njemu je u RFC4033), a podrazumijeva i korištenje EDNS0 podrške iz RFC2671. Dodatni RR-ovi koje DNSSEC donosi su dokumentirani u RFC4034. Dodatne modifikacije DNS protokola poradi ekstenzija su dokumentirane u RFC4035.

- TSIG (eng. *Transaction Signature*) - omogućava jednostavnu autentifikaciju koristeći dijeljene tajne ključeve i hashiranje za DNS transakcije,
- RP (eng. *Responsible Person*) - zapis o odgovornoj osobi za domenu ili čvorove.

Postoji još niz rijetko korištenih zapisa [8]: AFSDB (eng. *AFS Database Location Code*), HINFO (eng. *Host Information*), ISDN (eng. *ISDN Address*), MB (eng. *Mailbox*), MR (eng. *Mail Rename Domain Code*), NULL (eng. *Null Record*), RT (eng. *Route Through*), X25 (eng. *X25 PSDN Address*), MINFO (eng. *Mailbox or Mailing List Information*), PX (eng. *Pointer to X.400/RFC822 Mail Mapping Information*), NSAP (eng. *Network Service Access Point Address*) i NAPTR (eng. *Naming Authority Pointer*).

2.4. DNS upiti i odgovori

Standardni DNS upit je obično vrlo jednostavan i sadrži uglavnom samo podatak koji se želi razriješiti, dok su odgovori uvijek složeniji budući da sadrže sve adrese i zamjenske adrese koje su rezultat upita. Stoga se odgovori obično sažimaju posebnim algoritmima[4], eliminirajući nepotrebne podatke [9] i smanjujući samu veličinu UDP datagrama. U slučaju da veličina paketa i dalje prelazi 512 bajtova, šalje se parcijalna poruka u obliku UDP paketa s postavljenim posebnim bitom koji označuje da se upit mora ponoviti koristeći TCP. Navedena maksimalna veličina paketa je ujedno i razlog zašto postoji svega 13 vršnih DNS poslužitelja: upravo se lista od samo 13 IP adresa može spremiti u jedan DNS paket. Tipičan izgled DNS paketa prikazan je na slici 2.3, a podrobnije se definira u nastavku.



Slika 2.3: Prikaz DNS paketa s odjeljcima

Za upite i odgovore se koristi tzv. opći oblik poruke, koji se sastoji od 5 odjeljaka prikazanih u tablici 2.1. Dotična poruka se popunjava upitom od klijenta i odgovorom od poslužitelja, te u oba slučaja i podacima u zaglavlju koji su nužni da se proces obavi ispravno i uspješno.

Tablica 2.1: Odjeljci u DNS paketu

naziv odjeljka	**svrha odjeljka**
Zaglavlje (eng.	Nužna polja koja definiraju tip poruke i pružaju klijentu ili poslužitelju važne

4 Više o kompresiji imena u DNS paketima je moguće pročitati u RFC1035, poglavlju 4.1.4.

naziv odjeljka	svrha odjeljka
Header)	informacije o poruci. U zaglavlju se također 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.
Odjeljak pitanja (eng. *Question Section*)	Sadrži jedan ili više upita klijenta prema DNS poslužitelju. Veličina ovisi o broju upita (iako je najčešće samo jedan upit).
Odjeljak za odgovor (eng. *Answer Section*)	Sadrži jedan ili više RR-ova koji su odgovor na klijentov upit. Veličina ovisi o broju odgovora.
Autoritativni odjeljak (eng. *Autority Section*)	Sadrži 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. Veličina ovisi o broju autoritativnih zapisa.
Dodatni odjeljak (eng. *Additional Section*)	Sadrži 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. Veličina ovisi o broju dodatnih zapisa.

Svaka DNS poruka (upit ili odgovor) ima nekoliko polja u zaglavlju koja definiraju najvažnije karakteristike poruke. Tablica 2.2 prikazuje strukturu zaglavlja, odnosno polja zajedno s njihovim veličinama [10].

Tablica 2.2: Prikaz zaglavlja u DNS paketu

naziv polja	veličina	opis
ID (eng. *Identifier*)	2 bajta	16-bitni identifikator paketa koji se iz upita prenosi u odgovor, te se na taj način povezuje upit i odgovor. Jedinstven je za pojedini poslužitelj u kontekstu pojedine komunikacije te se generira više ili manje pseudoslučajno ovisno o DNS poslužitelju.
QR (eng. *Query/Response Flag*)	1 bit	QR=0 za upit prema poslužitelju, odnosno QR=1 za odgovor od poslužitelja.
OPCODE (eng. *Operation Code*)	4 bita	Definira tip upita. Vrijednosti su sljedeće: • OPCODE=0 je QUERY, uobičajeni tip upita, • OPCODE=1 je IQUERY, inverzni upit koji se danas više ne koristi, • OPCODE=2 je STATUS, upit za doznavanje stanja poslužitelja, • OPCODE=3 se ne koristi, • OPCODE=4 je NOTIFY, specijalna poruka koja se koristi za obavijest poslužitelju kako su se podaci u pojedinoj zoni za domenu promijenili, te da je potrebno obaviti prijenos zone, • OPCODE=5 je UPDATE, specijalna poruka koja služi za implementiranje dinamičkog DNS-a, odnosno načina za dodavanje, izmjenu i brisanje zapisa.
AA (eng. *Authoritative*	1 bit	AA=1 u odgovoru označava da je poslužitelj autoritativan za zonu u odjeljku za pitanja, odnosno AA=0 znači da odgovor nije autoritativan što je

naziv polja	veličina	opis
Answer Flag)		karakteristično za DNS rekursore.
TC (eng. *Truncation Flag*)	1 bit	TC=1 u odgovoru označava da bi puni UDP odgovor bio veći od 512 bajtova, te da je potrebno prijeći na TCP komunikaciju te ponoviti upit. Dobiveni UDP odgovor sadrži parcijalni dio traženih informacija.
RD (eng. *Recursion Desired*)	1 bit	RD=1 u upitu označava da klijent traži rekurziju. Eventualni odgovori zadržavaju stanje zastavice.
RA (eng. *Recursion Available*)	1 bit	RA=1 u odgovoru označava da poslužitelj podržava rekurziju. Isključivo autoritativni poslužitelj neće podržavati rekurziju.
Z (eng. *Zero*)	3 bita	Rezervirano, trebaju biti 0.
RCODE (eng. *Response Code*)	4 bita	Definira rezultat obrade upita. Vrijednosti su sljedeće: • RCODE=0 je u svim pitanjima, kao i u odgovorima koji nisu rezultirali pogreškom (*No Error*), • RCODE=1 kad postoji pogreška u formatu upita (*Format Error*), • RCODE=2 kad poslužitelj nije u mogućnosti odgovoriti zbog unutrašnje pogreške (*Server Error*), • RCODE=3 kad ime navedeno u upitu nije nađeno u domeni (*Name Error*). Odgovor može biti autoritativan ili neautoritativan (npr. negativni DNS međuspremnik), • RCODE=4 kad tip upita nije podržan od strane poslužitelja (*Not Implemented*), • RCODE=5 kad poslužitelj odbija obaviti upit, primjerice zbog pristupnih listi s obzirom na tip upita (*Refused*), • RCODE=6 kad traženo ime postoji, a ne bi smjelo (*YX Domain*), • RCODE=7 kad traženi zapis postoji, a ne bi smio (*YX RR Set*), • RCODE=8 kad traženi zapis ne postoji, a trebao bi (*NX RR Set*), • RCODE=9 kad poslužitelj nije autoritativan za traženu domenu (*Not Auth*), • RCODE=10 kad traženo ime nije unutar zone iz poruke (*Not Zone*).
QDCOUNT (eng. *Question Count*)	2 bajta	Određuje broj upita u odjeljku za pitanja. Upit ima svega jedno pitanje, pa se obrada višestrukih pitanja razlikuje između različitih poslužiteljskih softvera.
ANCOUNT (eng. *Answer Record Count*)	2 bajta	Određuje broj RR-ova u odjeljku za odgovore.
NSCOUNT (eng. *Authority Record Count*)	2 bajta	Određuje broj RR-ova u autoritativnom odjeljku.
ARCOUNT (eng. *Additional Record Count*)	2 bajta	Određuje broj RR-ova u dodatnom odjeljku.

Tablica 2.3 prikazuje koja polja ima odjeljak upita u DNS paketima, te koje su njihove veličine.

Tablica 2.3: Polja u odjeljku upita

naziv polja	veličina	opis
QNAME (eng. *Question Name*)	varira	Sadrži objekt, domenu ili zonu koji su predmet upita.
QTYPE (eng. *Question Type*)	2 bajta	Sadrži tip upita. Može sadržavati specifični broj koji odgovara tipu RR-a koji se traži ili pak neki od posebnih brojeva za posebne vrste upita: • QTYPE=251 odgovara zahtjevu za inkrementalni zonski prijenos (IXFR) • QTYPE=252 odgovara standardnom zahtjevu za prijenos zone (AXFR) • QTYPE=253, QTYPE=254 odgovaraju zastarjelim upitima za zapise vezane uz e-mail (MAILA i MAILB upiti za MB, MG i MR zapisima), • QTYPE=255 koji odgovara upitu za svim zapisima ("*").
QCLASS (eng. *Question Class*)	2 bajta	Označava koji se tip RR traži i može poprimiti vrijednost od 0 do 65535. Standardno se koristi svega pet vrijednosti: • QCLASS=1 za Internet (IN) zapis, • QCLASS=3 za CHAOS, • QCLASS=4 za Hesiod (HS), • QCLASS=254 za prazni (NONE) tip koji se obično koristi u dinamičkom DNS-u, • QCLASS=255 za ANY upit. ANY klasa je zamjenski ("*") tip.

2.5. Tipovi DNS poslužitelja

Potrebno je još definirati i međuodnos više DNS poslužitelja za istu domenu. 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. Riječ je o poslužitelju koji servira vlastite podatke klijentima, a on 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 prihvat 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 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. Naime, 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), kad je on neaktivan ili neispravan u dužem periodu (veći od TTL-a) navedena zona će biti nedostupna. Tim više, nakon pokretanja poslužitelja zona će biti još uvijek nedostupna s obzirom da se neuspjeli upiti (oni koji su dobili NXDOMAIN za odgovor) pamte još neko vrijeme na klijentima i poslužiteljima zbog principa negativnog međuspremnika. Stoga je razvijen princip primarnog (eng. *Primary/Master*) i sekundarnog (eng. *Secondary/Slave*) DNS poslužitelja.

Primarni poslužitelj je onaj autoritativni poslužitelj koji podatke o svojoj zoni ima lokalno spremljene, odnosno ima lokalni pristup navedenim podacima. Sekundarni poslužitelj je pak onaj koji dobiva podatke od nekog vanjskog izvora, obično koristeći prijenos zone (eng. *Zone Transfer*) od primarnog poslužitelja. Primarni poslužitelj za jednu zonu može biti sekundarni za drugu i sl. S gledišta klijenta, oba su poslužitelja (primarni i sekundarni) jednake vrijednosti (autoriteta) i jednakog prioriteta (slučajni izbor). Postoje i drugi razlozi za uvođenje 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.

Osim primarnih i sekundarnih autoritativnih poslužitelja postoji još par tipova poslužitelja. Prvi u nizu je isključivo međuspremnički poslužitelj (eng. *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 pamćenje rezultata DNS upita, smanjujući time opterećenje na autoritativnim poslužiteljima. Sljedeći tip je prosljeđivački poslužitelj (eng. *Forwarding Name Server*). Njegova je osnovna funkcija prihvat i prosljeđivanje upita nekom drugom DNS poslužitelju, ali se obično kombinira i s lokalnom pohranom dobivenih rezultata, pa je riječ o dobrom rješenju za spore mreže.

Sljedeći tip je isključivo autoritativni poslužitelj (eng. *Authoritative-only Name Server*) koji nema međuspremnik DNS upita niti ne odgovara na upite za koje nije autoritativan. On je 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. Takve okoline gdje se traži sigurni oblik DNS poslužitelja obično imaju nekoliko DNS poslužitelja od kojih su samo neki javno vidljivi, dok su drugi skriveni (eng. *Stealth Name Server*). Najčešće je slučaj da skriveni poslužitelji isporučuju klijentima DNS informacije koje nisu vidljive na javnoj vanjskoj mreži. Na taj se način vanjskim klijentima poslužuje tek dio informacija za koje se smatra da su im potrebne, a unutrašnjima se daje drugi dio informacija - za koji se smatra da su im dovoljne - i tako se eliminira sigurnosni problem da svi vide "sve". Taj princip se još naziva

razdvojeni poslužitelji (eng. *Split Name Server*), odnosno razdvojeni DNS (eng. *Split DNS*).

2.6. Sigurnosni problemi

Postoji niz trikova pomoću kojih se može odredišni DNS poslužitelj natjerati da prihvati lažne zapise[5]. Takvom se metodom lažiranja DNS zapisa (eng. *DNS Forgery*) [11] nesvjesni klijenti preusmjeruju na lažne adrese i time postaju laka meta napadača. Standardno su takvi napadi u formi trovanja DNS međuspremnika (eng. *Cache Poisoning*) [13], napada kod kojeg se utiče na DNS poslužitelj da povjeruje da je dobio autoritativne informacije o traženim podacima. 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.

Drugi veliki problem nije toliko vezan uz sigurnost koliko uz DNS zagađenje (eng. *DNS Pollution*) [12] odnosno bespotrebne DNS upite. Tipični primjer ovakvog prometa su DNS upiti za privatnim adresama[6] koje je potrebno lokalno razriješiti na DNS poslužitelju tako da se ne prosljeđuju dalje. Takav promet bespotrebno opterećuje vršne DNS poslužitelje budući da se takve adrese koriste isključivo u privatnim mrežama, te niti jedan DNS poslužitelj u svijetu neće biti autoritativan za navedene adrese. Prema recentnim istraživanjima čak 1.61% ukupnog svjetskog DNS prometa predstavlja curenje RFC1918 upita prema vršnim DNS poslužiteljima [13], stoga je 2002. godine formirana dodatna usmjerivačka hijerarhija oko AS112 (eng. *Autonomous system*) radi razrješavanja upita za RFC1918 (10.in-addr.arpa, itd.) i RFC3330 (254.169.in-addr.arpa) adresama. Ono što se može zaključiti jest da je relativno malen postotak DNS prometa u stvari korektan (prema dosadašnjim mjerenjima na vršnim poslužiteljima, svega 2% [14]).

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 s IP adresom 1.2.3.4?"). Ovo je karakteristično za Microsoft Windows NT operacijski sustav, a rješava se obično korištenjem djbdns servisa ili Bind 9 poslužitelja koji je autoritativan za svih 256 numeričkih zona, pri čemu je svaka prazna,
- Upiti za krivim TLD-ovima - koji su najčešće pogreška u lokalnim konfiguracijama (kriva domena, netočna domena, mobilni uređaji, neispravne standardne konfiguracije) ili aplikacijama, pa se pojavljuju upiti za "localhost", "localdomain", "workgroup" i sličnim nepostojećim domenama, odnosno domenama koje bi trebale biti lokalno definirane,

5 Analiza potencijalnih sigurnosnih problema u DNS sustavu je dokumentirana u RFC3833.

6 Misli se na RFC1918 privatne adrese, odnosno 10/8, 172.16/12 i 192.168/16 prefikse.

- Upiti za adresama vršnih poslužitelja - svi DNS poslužitelji imaju popis vršnih poslužitelja kako bi uopće mogli ostvariti početnu komunikaciju. Povremeno osvježavanje zapisa je normalno zbog istjecanja TTL-a, no RR-ovi za vršne poslužitelje imaju najčešće TTL od 1000 sati. U slučaju da se ovakvi upiti dešavaju prečesto, riječ je o pogrešci u filtriranju DNS prometa, neispravnom DNS poslužiteljskom softveru i sl,
- IPv6 upiti - često ih šalju aplikacije kad to nije nužno. Bind poslužitelj dodatno obavlja najčešće nepotrebne (čak i ako računalo nema IPv6 stog) AAAA i A6 upite, primjerice za povezujuće zapise.

Iz priloženog je očito da bi DNS administratoru bio krajnje koristan sustav koji prati dolazni i odlazni DNS promet te identificira potencijalne sigurnosne prijetnje, prijavljuje i eventualno obavlja proaktivne radnje zaštite od istih. U nastavku će se ustvrditi postoje li takvi alati i koje su njihove funkcionalnosti.

2.7. Metode analize prometa u poslužiteljima

Svaki rekurzivni DNS poslužitelj mora implementirati temeljitu analizu odgovora s udaljenog poslužitelja radi izbjegavanja trovanja i radi provjere korektnosti odgovora s obzirom na implementirane DNS standarde. Temeljitost takve provjere se bitno razlikuje od poslužitelja do poslužitelja, kako implementacijski tako i kvalitetom dobivenih rezultata.

Tijekom istraživanja se pokazalo da su postojeće metode bilježenja problema uočenih u DNS prometu uglavnom nepotpune i neprilagođene eventualnoj daljnjoj sigurnosnoj analizi. Primjerice najčešće nedostaje detaljni prikaz primljenog DNS paketa (izvorišna i odradišna adresa, zastavice na svakom pojedinom nivou paketa, dužine pojedinog zaglavlja, detaljni sadržaj podatkovnog dijela paketa, itd.), razlog odbacivanja pojedinog upita, kao i popis paketa koji su odbačeni jer bi doveli do sigurnosnog incidenta (npr. u slučaju pokušaja trovanja DNS spremnika). Niti kod jednog nije moguće slanje zapisnika prema udaljenom poslužitelju za bilježenje, već je to samo kod nekolicine moguće indirektno ostvariti koristeći Syslog mehanizam. I kod takvog korištenja problem je da Syslog ne podržava kriptiranje prometa i odgovarajuću autentikaciju/autorizaciju, već je radi toga nužno raditi sistemske zahvate u vidu IPsec (eng. *Internet Protocol Security*) tunela.

Tablica 2.4 donosi najpoznatije DNS poslužitelje koji su pregledani tijekom istraživanja i njihove pojedinačne karakteristike [15] na poljima analize potencijalno problematičnih DNS upita, bilježenja problema u sistemske zapisnike, bilježenja dolaznih i odlaznih DNS paketa, kao i potpunost implementacije DNS standarda.

Tablica 2.4: Pregled analize prometa u DNS poslužiteljima

naziv poslužitelja	bilježenje i analiza prometa
Bind	Bilježenje je moguće, nije standardno aktivirano ali se jednostavno konfigurira. Vide se samo opći detalji o upitu (samo upitu, nije moguće vidjeti odgovore), ne i specifične tehničke pojedinosti (broj odjeljaka, veličina paketa, pogreške u upitu ili odgovoru, status upita, itd). Postoji sigurnosno logiranje no ono je prvenstveno orijentirano lokalnoj konfiguraciji poslužitelja (pristupne liste, upiti koje je nemoguće riješiti, nesigurni dinamički DNS upiti, prijenos zone, itd). Sama klasifikacija bilježenja je dobro razrađena. Kao i kod drugih poslužitelja nije moguće unutrašnje udaljeno logiranje, već jedino kroz Syslog mehanizam. Riječ je o jednom od najpotpunijih implementacija DNS poslužitelja, a analiza upita je vrlo složeno izvedena kroz temeljitu iterativnu provjeru svakog pojedinog upita i odgovora. Poslužitelj kao i većina ostalih nema nikakve metode proaktivne zaštite u smislu reakcije na pokušaje napada. Bind je danas praktično standardni Unix/Linux DNS poslužitelj. URL: https://www.isc.org/software/bind
djbdns	Standardno bilježi promet i različite unutrašnje statistike vrlo detaljno, no dokumentacija o tome je uglavnom slaba ili nikakva. Nema mogućnost bilježenja poslanih odgovora niti eventualne sigurnosne incidente. U slučaju sigurnosnih implikacija ne šalje nikakav odgovor, ni to uglavnom ne bilježi osim u nekoliko slučajeva. Nije moguće ni logiranje kroz syslog niti udaljeno logiranje, jedino kroz eventualne nestandardne dodatke. Zabilježeni promet nije "ljudski" čitljiv (IP adrese u heksadecimalnom zapisu, numerički zapisane pogreške bez tablice značenja, itd). Autoritativni i rekurzivni poslužitelj su odvojeni, pa je uobičajeno razdvojeno bilježenje. Riječ je o osnovnoj DNS implementaciji, a softver uglavnom minimalno analizira DNS promet i donosi zaključke o njemu, već odgovara vrlo konzervativno i to samo na promet na koji može u potpunosti odgovoriti. Djbdns je vjerojatno trenutno najsigurniji DNS poslužitelj u smislu mogućih sigurnosnih implikacija zbog svog minimalističkog dizajna. URL: http://cr.yp.to/djbdns.html
MaraDNS	Standardno ne bilježi promet, međutim je to moguće aktivirati. Može bilježiti upite vrlo detaljno kao i sve moguće ustanovljene sigurnosne probleme, te bilo kakve probleme otkrivene u upitu od klijenta ili odgovoru od drugog poslužitelja. Nema klasifikaciju zabilježenih problema. Implementacija trenutnih DNS standarda je prilično potpuna, a autor je koristio ideje i iz djbdns i Bind softvera u analizi prometa. Provjera prometa je ostvarena uglavnom kao niz logičkih provjeri tijekom obrade prometa, što je manje formalan način i ostavlja mogućnost propusta u otkrivanju. URL: http://www.maradns.org/
PowerDNS	Standardno ne bilježi promet, no moguće je dobiti krajnje detaljno praćenje upita i odgovora kao i uočenih sigurnosnih problema. Standardno daje detaljniji i upotrebljiviji ispis od Bind poslužitelja, zajedno s različitim korisnim podacima o uočenim nepravilnostima u DNS prometu. Bilježenje je moguće uspostaviti i kroz Syslog servis.

naziv poslužitelja	bilježenje i analiza prometa
	Što se tiče DNS standarda, riječ je o jednom od najpotpunijih DNS poslužitelja uz Bind. Provjera prometa je vrlo detaljna kao i bilježenje problematičnog prometa. Softver u slučaju opetovanih pokušaja trovanja ili DoS napada reagira progresivnim blokiranjem komunikacije prema napadaču. URL: http://www.powerdns.com/
NSD	Riječ je o isključivo autoritativnom DNS poslužitelju, pa je i sama analiza mogućih prijetnji bitno jednostavnija nego kod ostalih poslužitelja. Nema bilježenja upita i odgovora, već bilježenje eventualnih funkcionalnih problema i zabranjenih upita zbog pristupnih listi. URL: http://www.nlnetlabs.nl/projects/nsd/
Microsoft DNS	Standardno ne bilježi promet, ali je moguće bilježiti i dolazni i odlazni promet, što je rijetka mogućnost. Postoji upotrebljiva klasifikacija problema, iako je sigurnosno bilježenje uglavnom dosta ograničene funkcionalnosti. Implementirani su svi važniji DNS standardi. Nepoznato je koliko je dobro ostvarena provjera prometa, no poslužitelj je relativno dugo bio ranjiv na različite varijante DNS trovanja. Sam poslužitelj je praktički osnovni dio Microsoft Active Domaina arhitekture, pa je shodno tome i vrlo popularno rješenje. URL: http://www.microsoft.com
Dnsmasq	Riječ je o prosljeđivačkom tipu DNS poslužitelja koji može eventualno autoritativno davati podatke o lokalnim računalima. Ne bilježi promet niti ima takve mogućnosti, s obzirom da je primarno namijenjen za lokalni rad. Sigurnosna analiza i detaljna provjera DNS paketa je nepostojeća. URL: http://www.thekelleys.org.uk/dnsmasq/doc.html
Posadis	Nedostaje dokumentacija te nema više nikakve vidljive aktivnosti na razvoju softvera. DNS poslužitelj se barem 4 godine ne razvija, te je time izvan fokusa ovog pregleda. Ranjiv je na cijeli niz novijih sigurnosnih napada. URL: http://posadis.sourceforge.net/
Unbound	Standardno ne bilježi promet, ali je to moguće ostvariti za dolazni promet. Bilježenje sigurnosnih incidenata je nedovoljno detaljno, iako postoji. Otkrivanje potencijalnih problema je ostvareno vrlo detaljno i kvalitetno, izgradnjom potpunog DNS stabla i odbacivanjem svih dijelova paketa koji predstavljaju suvišak ili potencijalni problem, zajedno s bilježenjem uočenih problema. Poslužitelj implementira sve bitne standarde i dio dodatnih DNS ekstenzija. Provjera prometa je kvalitetna i detaljnija od svih ostalih poslužitelja, provjerava se ne samo sukladnost standardima već i svaki potencijalni problem i to na način da se uzima u obzir samo korektne informacije. URL: http://www.unbound.net/
Simple DNS Plus	Moguće je bilježiti i dolazni i odlazni promet, kako lokalno tako i udaljeno, zajedno s bilježenjem sirovih (neobrađenih i potpunih) DNS paketa, čime se razlikuje od ostalih DNS poslužitelja. Ima relativno skromnu sigurnosnu analizu prometa, te dosta slabu klasifikaciju uočenih problema. Poslužitelj poštuje sve bitne standarde i dio dodatnih DNS ekstenzija. Što se tiče

naziv poslužitelja	bilježenje i analiza prometa
	ostalih mogućnosti nije na razini ostalih poslužitelja. URL: http://www.simpledns.com/
CNS, ANS, Secure64 DNS Signer, Secure64 DNS Authority	Komercijalni poslužitelji zatvorenog izvornog koda koji ne postoje u demo/trial varijantama, tako da se o njima ne može ustanoviti više konkretnih informacija. URL: http://www.nominum.com/, http://www.secure64.com/

2.8. Pregled postojećih specijaliziranih alata

U tablici 2.5 slijedi pregled specijaliziranih alata za analizu i obradu DNS prometa, te otkrivanje sigurnosnih problema zajedno s eventualnim mogućnostima bilježenja DNS prometa.

Tablica 2.5: Pregled alata za DNS analizu

naziv alata	bilježenje i analiza prometa
Snort IDS	Tipični Snort uzorci ne omogućavaju kvalitetno praćenje DNS prometa, no postoji dodatni dns_state alat koji služi otkrivanju nekoliko DNS ranjivosti. Može otkriti pokušaje trovanja (samo varijanta trovanja kroz iteriranje ID-jeva (eng. *Identification*) s lažnim odgovorima), fast-flux napade, odgovore s krivim ID-jevima, odgovore bez prethodnog odgovarajućeg upita kao i višestruke (nepotrebne) odgovore. Forenzička komponenta je kvalitetna, te je moguće snimati pakete sa svim mogućim zastavicama kao i PCAP (eng. *Packet Capture*) datoteke za kasniju dodatnu analizu. Ne postoji dedicirana DNS analiza paketa, no to je moguće obavljati s dodatnim alatima poput Wireshark analizatora. URL: http://www.snort.org/
dnstop	Služi prvenstveno statističkoj analizi DNS upita, broju IPv4/IPv6 DNS paketa, TLD, SLD (eng. *Second-level domain*), 3LD (eng. *Third-level domain*) upita, broju A, PTR, CNAME i ostalih RR uočenih u upitima odnosno odgovorima, broju viđenih domena, itd. Također, alat prepoznaje osnovne tipove problema u DNS upitima poput RFC1918 PTR upita, A-za-A upita i nepoznatih TLD-ova u upitima. Forenzička komponenta je relativno slaba. Ne postoji mogućnost bilježenja prometa, iako je npr. moguće analizirati spremljene PCAP datoteke. Postoji skroman broj samih IDS (eng. *Intrusion detection system*) filtara, no i dalje se dobiva više informacija nego što prikazuju DNS poslužitelji. URL: http://dns.measurement-factory.com/tools/dnstop/

naziv alata	bilježenje i analiza prometa
dnspktflow	Koristi se za vizualizaciju prometa između DNS klijenata i poslužitelja, s opcionalnim prikazom sadržaja, zastavica, itd. Sam alat nije samostojeći, već koristi više vanjskih programa (Graphviz za iscrtavanje grafova te Wireshark za samu analizu DNS prometa). Program nema nikakvu forenzičku vrijednost, te nema nikakvih IDS filtara koji bi prikazivali eventualne uočene probleme u zaprimljenom prometu. URL: http://www.dnssec-tools.org/
Wireshark (packet-dns)	Ethereal odnosno Wireshark je najpoznatiji analizator mrežnog prometa. Podržava niz različitih protokola na višim i nižim razinama OSI (eng. *Open Systems Interconnection*) modela, a između ostaloga i DNS kroz unutrašnji packet-dns modul. Moguće je analizirati TCP i UDP DNS promet, prikazivati rezultate na konzoli te spremati zaprimljene pakete u PCAP datoteke. Implementirani su svi poznati DNS standardi (kao i standardi koji su tek u proceduri za odobrenje) za ispravno dekodiranje DNS paketa, pa je moguća analiza proizvoljnog DNS prometa te kompletni prikaz zastavica i sadržaja paketa. Program se najčešće koristi u forenzičke svrhe, no nema nikakvih IDS filtara koji bi prikazivali eventualne uočene probleme. URL: http://www.wireshark.org/
dnscap	Riječ je o alatu namijenjenom prisluškivanju i bilježenju DNS prometa. Program funkcionira kao prisluškivač prometa s mogućnostima vrlo detaljnog odabira željenih DNS paketa (tipovi upita odnosno odgovora, zastavice, itd.) te bilježenja u datoteke, bilo PCAP bilo prezentacijskog (već dekodiranog) tipa. Alat nema nikakvih IDS filtara, no ima sve potrebne mogućnosti za selekciju i bilježenje željenog prometa te kasniju analizu drugim alatima. URL: https://www.dns-oarc.net/tools/dnscap

3. Sustav za nadzor i analizu DNS prometa

S obzirom na spomenutu sigurnosnu problematiku, smatra se da je svrsishodno izraditi sustav koji omogućava snimanje DNS upita i odgovora, njihovu aplikativnu analizu te bilježenje rezultata analize kao i odgovarajuće prezentiranih paketa. Takav bi projekt bio krajnje koristan svim sistem-administratorima s obzirom da ne postoji rješenje (kao što je pokazano u poglavljima 2.7 i 2.8) koje bi arhitekturalno i funkcionalno omogućilo distribuiran prihvat DNS podataka, temeljitu sigurnosnu analizu i skladištenje prezentiranih podataka. U okviru ovog rada napraviti će se i analiza hrvatskog DNS prometa jednom od većih DNS poslužitelja.

Koristeći se višegodišnjim iskustvom administracije različitih Unix poslužitelja, moguće je definirati sljedeće zahtjeve koje se očekuje da aplikacija zadovolji:

- nužna je visoka razina modularnosti i buduće nadogradivosti, a podrazumijeva se korištenje gotovih standardiziranih komponenti (za mrežnu komunikaciju, obradu konfiguracijskih datoteka, primitive za međuprocesnu komunikaciju i zaključavanje, kriptografske funkcije kao i različite više tipove podataka poput listi, asocijativnih rječnika, itd.),
- traži se potpuna prenosivost između različitih modernih operacijskih sustava, bez potrebe za ponovnim prevođenjem ili korekcijama u izvornom kodu,
- prikupljanje DNS podataka ne smije niti u kojem obliku usporiti ili ometati normalan rad računala na kojem se prikupljaju i analiziraju DNS paketi,
- prikupljanje mora biti moguće ostvariti lokalno (isključivo lokalna analiza i spremište) i distribuirano (model jednog ili više agenata-senzora te udaljenog centralnog poslužitelja-spremišta),
- komunikacija agenata i centralnog poslužitelja mora biti izvedena tako da se postignu što manje latencije u komunikaciji (minimalno usporenje centralnog poslužitelja i agenata) te je nužno izbjeći blokiranje senzora ili centralnog poslužitelja zbog pogrešaka u komunikaciji,
- između agenata i udaljenog spremišta nužno je ostvariti odgovarajuću autorizaciju i autentikaciju, a podaci koji se prenose udaljenom spremištu moraju se kriptirati uobičajenim snažnim kriptografskim metodama (u slučaju isključivo lokalnog prikupljanja kriptirana komunikacija nije nužnost),
- aplikacija mora obavljati odgovarajuću aplikativnu analizu i dekodiranje primljenih paketa, utvrđujući IP izvorište, IP odredište, dodatne zastavice i oznake kao i sav sadržaj DNS poruka (također sa svim popratnim zastavicama i oznakama),
- nužno je omogućiti pozadinski (servisni) način rada gdje se aplikacije izvršavaju bez direktne komunikacije s grafičkim ili tekstualnim terminalom,

- za agente i spremište je potrebno ostvariti obradu svih pogrešaka tijekom rada uz nastavak rada ako je moguć, te odgovarajuće bilježenje navedenih pogrešaka u zapisnike.

Zbog navedenih zahtjeva smatra se da je optimalan izbor programski jezik Python. Navedimo nekoliko bitnih karakteristika navedenog jezika koje opravdavaju ovaj izbor:

- Python je moderni objektno-orijentirani interpretirani jezik (nije potrebna rekompilacija koda pri prenošenju na druge platforme) s nizom kvalitetnih implementacija (CPython, Jython, IronPython) na praktički svim modernim operacijskim sustavima,
- već postoji niz odgovarajućih modula koji jamče prenosivost i dovoljnu apstrakciju karakteristika pojedinog operacijskog sustava, tako da je moguće prvenstveno se koncentrirati na rješavanje traženih problema (DNS dekodiranje, analiza i prezentacija) ne gubeći se u implementacijskim detaljima,
- za Python postoji već odgovarajući model prihvata mrežnih paketa od operacijskog sustava: Scapy biblioteka omogućava odgovarajuću interaktivnu disekciju i manipulaciju paketima te podržava UDP DNS pakete; tijekom testiranja ustanovljeno je da nije u stanju dekodirati TCP DNS pakete no to je riješeno dodavanjem odgovarajućeg koda (koji je također dio ovog diplomskog rada).

Praktični rad bi se stoga temeljio na ostvarenju sljedećih funkcionalnih cjelina:

- minimalnog zahvata na Python Scapy biblioteci za podršku TCP DNS paketima,
- Python implementaciji samostojećeg agenta za prihvat DNS paketa, osnovno dekodiranje i odašiljanje kroz kriptiranu vezu prema udaljenom centralnom poslužitelju,
- Python implementaciji centralnog poslužitelja za prihvat dekodiranih paketa, višestruku sigurnosnu analizu i bilježenje u lokalne zapisnike.

3.1. Razrada implementacije

U nastavku će se opisati rješenja pojedinih problema iz prethodnog poglavlja, odnosno idejna i programska rješenja koja su predmet ovog diplomskog rada: kako je postignuta brzina i djelotvornost komunikacije, detalji o vlastitom komunikacijskom protokolu, detalji o zaštiti podataka te ostalim bitnim komponentama.

3.1.1. Efikasna komunikacija

S obzirom na tipični zahtjev za niskom latencijom pri ostvarenju komunikacijskog kanala, najjednostavnije i ujedno najefikasnije rješenje je koristiti jednosmjernu UDP komunikaciju. Time se izbjegavaju čekanja na ostvarenje trosmjernog rukovanja karakterističnog za TCP vezu, a postiže se i jednosmjernost komunikacije koja je sasvim dovoljna za ovaj slučaj komunikacije između više agenata i jednog centralnog spremišta. Sama vršna propusnost će biti nešto manja, no kako je riječ o relativno malim paketima (uglavnom ispod 1 kilobajta sadržaja) to ne predstavlja značajan problem.

Eventualni problem jest da je je izvorište UDP poruka lako lažirati (ne postoji implicitna provjera izvorišta s obzirom da se ne ostvaruje dvosmjerna komunikacija) pa je stoga potrebno implementirati mehanizam pristupnih listi kao i odgovarajuću kriptografsku zaštitu sadržaja. Odsutnost "stalnih" TCP konekcija na centralnom poslužitelju također pojednostavljuje implementaciju s obzirom da se promet obrađuje pojedinačno (po paketu) te da se ne gomilaju konekcije odnosno otvorene datoteke.

3.1.2. Minimalno opterećenje računala senzora

Učinkovita metoda minimizacije dodatnog opterećenja na poslužitelj jest implementacija asinhronog prihvata podataka odnosno asinkrone mrežne komunikacije s centralnim poslužiteljem. S obzirom da Scapy biblioteka pruža mogućnost korištenja definiranja proizvoljne callback metode za prihvat podataka (u našem slučaju DNS prometa), poželjno je što manje procesorskog vremena provesti u navedenoj metodi s obzirom da se navedena metoda poziva za svaki primljeni paket željenog protokola, a to se očito može desiti i tisućama puta u sekundi na opterećenom DNS poslužitelju.

Rješenje jest da se svaki prihvaćeni paket u callback metodi odmah sprema u odgovarajuću FIFO (eng. *First In, First Out*) strukturu odnosno odgovarajući red poruka iz kojeg će jedan po jedan vaditi i analizirati u odvojenoj dretvi koja ne utiče na prihvat DNS podataka. Za red poruka se podrazumijeva da mora implementirati i odgovarajuće zaključavanje prilikom ažuriranja unosa te blokiranje dretve koja čita prazan red - ovime se postiže da program ne koristi procesorsko vrijeme dok nema podataka za analizu. Mrežna komunikacija se može obavljati u istoj dretvi u kojoj se obavlja i analiza, s obzirom na postojanje reda poruka za prihvat prikupljenih između dva pozivanja navedene dretve.

3.1.3. Minimalno opterećenje centralnog poslužitelja

Glavni problem centralnog poslužitelja jest velik broj mrežnih komunikacija koje se odvijaju istovremeno. Ovaj se problem najčešće rješava tako da UDP komunikacijski poslužitelj stvara novu instancu za svaki prispjeli paket, te se paketi paralelizirano

obrađuju. Postoje su dvije mogućnosti za paralelizaciju: stvaranje cijelih novih procesa za svaki pojedini paket ili korištenje dretvi. Ukoliko se koriste dretve cijeli je sustav bitno brži s obzirom na vrlo djelotvorno stvaranje dretvi, budući da se za dretve se koristi COW (eng. *Copy-on-write*) princip, dok stvaranje procesa radi kompletnu kopiju adresnog prostora što je tipično za red veličine sporija operacija. Daljnja je optimizacija kratki život pojedine dretve i minimalni kod koji se paralelizirano izvršava za pojedini paket.

Shodno tome, UDP poslužitelj za svaki prispjeli paket treba stvoriti odgovarajuću dretvu gdje se dešavaju sve specifične provjere (dekriptiranje itd.) za pojedini paket, a onda se dekodirani paket sprema u red poruka i dretva završava s radom. Zasebna dretva u redovnim intervalima obrađuje već dekodirane i provjerene pakete iz reda poruka u potrazi za sigurnosnim problemima. Kako je taj potonji dio posla najsporiji i s gledišta resursa najzahtjevniji dio posla, resurse će minimalno potrošiti upravo navedeni oblik rješenja. Za red poruka se podrazumijeva da mora provesti i odgovarajuće zaključavanje prilikom ažuriranja unosa, s obzirom da će ga ažurirati velik broj dretvi koji prvenstveno ovisi o broju aktivnih senzora.

Inicijalnu obradu uhvaćenih DNS paketa je povoljnije ostaviti da se odradi na pojedinačnim senzorima, s obzirom da bi se u suprotnom slučaju ukupni posao na centraliziranom poslužitelju višestruko povećao ili čak doveo do preopterećenja poslužitelja u ovisnosti o ukupnom broju aktivnih senzora i uhvaćenom prometu. Stoga je povoljnije ako senzori centralnom poslužitelju šalju već dekodirane informacije u odgovarajućem prezentacijskom (obrađenom i povoljnom za slanje preko mreže) obliku.

3.1.4. Kriptiranje prometa i provjera autentičnosti

Kad bi prijenos DNS paketa između senzora i centralnog poslužitelja bio u obliku čistog teksta, to bi omogućilo trećoj strani barem pasivno neovlašteno praćenje tog prometa, ali i razne vrste sigurnosnih napada s lažiranjem prometa (lažiranje zapisa, promjena u paketima ili napadi uskratom usluge). Stoga je nužno primijeniti odgovarajuće kriptografske sigurnosne mjere kako bi se zaštitilo od takvih napada. Odgovarajuće rješenje koje se temelji na podršci od operativnog sustava bilo bi korištenje IPsec[7] protokola, međutim ne podržavaju svi operativni sustavi IPsec ili integraciju s istim kroz IKEv2 (eng. *Internet Key Exchange*).

Problem je prikladno riješiti na razini same aplikacije i to tako da se prihvaćeni DNS paketi kriptiraju dodajući im zaštitnu sumu na agentu, dok se na centralnom poslužitelju promet provjerava i tek onda dekriptira. S obzirom da se ključevi unaprijed mogu rasporediti po svim poznatim senzorima i centralnom poslužitelju (tehnika koja se naziva PSK odnosno eng. *Pre-shared key*), može se koristiti simetrična enkripcija s istim ključem

7 IPsec je pobliže definiran u RFC4301. Također bitni standardi za proučiti su IPsec AH iz RFC4302, IPsec ESP iz RFC4303 te IKEv2 iz RFC4306.

za sve poslužitelje. Simetrični su algoritmi tipično za par redova veličina brži od asimetričnih, a koriste i znatno manje resursa tijekom rada.

Jedan od najpopularnijih simetričnih blokovskih algoritama i tipično u upotrebi u IPsec komunikaciji je AES (eng. *Advanced Encryption Standard*). Njegove prednosti su laka softverska implementacija, mala potrošnja resursa (s obzirom da je algoritam vrlo efikasan i supstitucijskog tipa, za svoje potrebe koristi vrlo malo radne memorije), otvorenost, standardiziranost, jednostavnost korištenja kao i niz gotovih implementacija za sve jezike pa tako i za Python.

AES radi na blokovima od 128 bitova i koristi ključeve od 128, 192 ili 256 bitova. Ima 5 načina rada: CBC (eng. *Cipher Block Chaining*), ECB (eng. *Electronic CodeBook*), CFB (eng. *Cipher FeedBack*), OFB (eng. *Output FeedBack*) and CTR (eng. *Counter*). Najpopularniji način koji je ujedno i najdetaljnije dokumentiran jest CBC, za koji je nužno koristiti inicijalizacijski vektor (IV odnosno eng. *Initialization vector*) koji je iste veličine kao i blok podataka. Navedeni vektor se prvo puni slučajnim brojevima, a zatim se obavlja XOR operacija s prvim blokom podataka prije enkriptiranja. Zatim se svaki sljedeći blok XOR-a s prethodnim nekriptiranim blokom prije enkripcije. Da bi druga strana mogla dekriptirati sadržaj, potrebno joj je prenijeti enkriptirane blokove kao i sam IV.

S obzirom da AES radi isključivo na blokovima podataka, nužno je osigurati da su dijeljeni ključ i podaci u odgovarajućim blokovima od 128 bitova, odnosno da je duljina ključa te poruke višekratnik veličine bloka. U slučaju da je blok nije potpuno popunjen, koristi se metoda popunjavanja po izboru. Tipično se u ovakvim slučajevima koristi PKCS#7[8] metoda koja popunjava ostatak bloka s brojem bajtova koji su dodani: za N bajtova koji su dodani do granice bloka, stavlja vrijednost N u svaki bajt koji je dodan.

Drugi bitan problem jest pravovremeno otkrivanje namjernih ili slučajnih modifikacija sadržaja poruke koristeći ICV (eng. *Integrity Check Value*) tehniku. Riječ je o izračunu jedinstvene vrijednosti za svaku poslanu poruku (poruka mora sadržavati IV i same enkriptirane podatke), ali takve vrijednosti koju nije moguće zlonamjerno promijeniti bez znanja tajnog ključa. U IPsec standardu se koristi RFC2104 HMAC (eng. *keyed-Hash Message Authentication Code*) kod kojeg se vrijednost derivira iz poruke i tajnog (dijeljenog) ključa. HMAC se istovremeno koristi kako za autentikaciju podataka, tako i za provjeru samog integriteta. HMAC omogućava korištenje nekih od tipičnih hash algoritama poput RFC1321 MD5 (eng. *Message-Digest Algorithm 5*) ili RFC3174 SHA1 (eng. *Secure Hash Algorithm 1*), a snaga zaštite direktno ovisi o kriptografskoj funkciji na kojoj se temelji. Kako je SHA1 je kriptografski značajno jača metoda koja proizvodi 160-bitnu sumu, HMAC-SHA1 se postavlja kao bolji izbor.

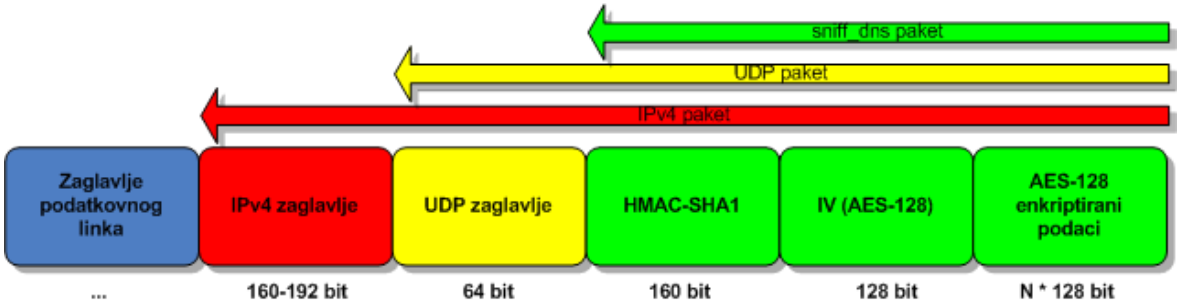
Svaki paket koji se šalje od agenta odnosno senzora prema centralnom poslužitelju ima redom sljedeća polja (ne navodimo IP zaglavlje, već samo sadržaj):

- HMAC-SHA1 zaštitnu sumu dužine 20 bajtova (160 bitova),

8 PKCS#7 metoda je opisana u RFC3852, poglavlju 6.3.

- IV incijalizacijski vektor dužine 16 bajtova (128 bitova, s obzirom da se koristi AES-128-CBC),
- N blokova AES-128-CBC šifriranih podataka od kojih je svaki dužine 16 bajtova (N*128 bitova).

Iz navedenog proizlazi da je minimalna duljina aplikativnog paketa 52 bajta, odnosno da je minimalna veličina IPv4 paketa 80 bajta. Slika 3.1 predočuje veličine zaglavlja kako osnovnog aplikativnog protokola tako i veličine zaglavlja ostalih razina niže komunikacije.



Slika 3.1: Pregled komunikacijskog paketa

U Pythonu je standardno implementiran HMAC (SHA1 i MD5 varijante), dok se za AES enkripciju koristi dodatni Crypto modul.

3.1.5. Autentikacija i autorizacija

Kao i kod svakog mrežnog servisa, nužno je moći ograničiti koji klijenti smiju komunicirati s centralnim poslužiteljem. U ovom sustavu je tu funkciju moguće provesti kroz dijeljeni ključ; s obzirom da je simetričan, svi klijenti koji imaju ključ mogu komunicirati s centralnim poslužiteljem. Druge mogućnosti su dodatno filtriranje po IP adresama, rasponima i sl. Međutim s gledišta brzine i sigurnosti takva filtriranja je daleko najefikasnije implementirati na vatrozidnom softveru, odnosno u jezgri poslužitelja.

Kada bi program i sadržavao individualne provjere IP izvorišta, to bi značilo da za eventualne lažirane pakete mora stvarati nove dretve u kojima obavlja sve sigurnosne provjere. To širom otvara mogućnost sigurnosnog napada uskratom rada, s obzirom da napadač može praktički neograničeno stvarati nova lažirana izvorišta i time opterećivati centralni poslužitelj do prestanka normalnog rada. Ako su pak sigurnosne provjere direktno implementirane u jezgri operacijskog sustava, to znači da eventualni zlonamjerni promet ni ne dolazi do same aplikacije i time minimalno utječe na normalnu komunikaciju i normalan rad poslužitelja.

3.1.6. Prijenos programskih struktura

Prije slanja programskih struktura iz memorije kroz mrežu, nužno ih je serijalizirati. Taj proces podrazumijeva konverziju objekata u niz bitova koji se može pohraniti na kakav medij ili poslati kroz mrežnu komunikaciju te uspješno kasnije učitati. Ta procedura stvara novi ali semantički isti objekt, kod kojeg su unutrašnje adrese različite, ali reference i sadržaj očuvani. Prije enkripcije podataka se unutrašnje memorijske strukture koje sadrže dekodirane DNS pakete moraju serijalizirati; odnosno, pri uspješnom zaprimanju enkriptiranih podataka se oni nakon provjera i dekripcije moraju deserijalizirati, stvarajući objekt s prezentiranim DNS podacima u memoriji.

Python sadrži odgovarajući Pickle/cPickle modul za navedene potrebe. Nadalje, Pickle objekti su sigurno kompatibilni između različitih verzija Python interpretera, što praktično znači da nisu nužne iste verzije Python programa na poslužitelju i na senzorima. Kompletne strukture u memoriji se pretvaraju u binarni (ili čisto tekstualni, ovisno o konfiguraciji) niz znakova koji je posve prenosiv bez ikakvih restrikcija.

3.2. Komponente i karakteristike sustava

Implementirani su u potpunosti sljedeći osnovni dijelovi sustava:

- senzor koji omogućava pasivno prisluškivanje proizvoljnog unicast i multicast DNS prometa u TCP i UDP obliku, osnovnu obradu podataka i njihovo kriptirano slanje prema centralnom poslužitelju,
- centralni poslužitelj koji prima podatke od jednog ili više senzora, provjerava, dekriptira i daljnje obrađuje, lokalno spremajući rezultate,
- komponenta za detaljnu sigurnosnu analizu koja omogućava prepoznavanje 12 tipova sigurnosnih napada; moguće ju je koristiti u svakom senzoru (za samostojeći rad) ili u centralnom poslužitelju (obrada svih prispjelih informacija),
- jednostavni DNS međuspremnik koji omogućava kontekstualnu analizu upita i odgovora.

Također je osmišljen i razvijen vlastiti protokol za kriptiranu komunikaciju sa zaštitnim sumama koji zadovoljava zahtjeve niskih latencija i mogućnosti prijenosa memorijskih struktura preko mreže.

U do sada opisanom sustavu djeluju sljedeći fizički uređaji:

- radne stanice, poslužitelji i sl. koji obavljaju DNS upite prema svojim DNS poslužiteljima na kojima su instalirani senzori za prihvat DNS prometa,

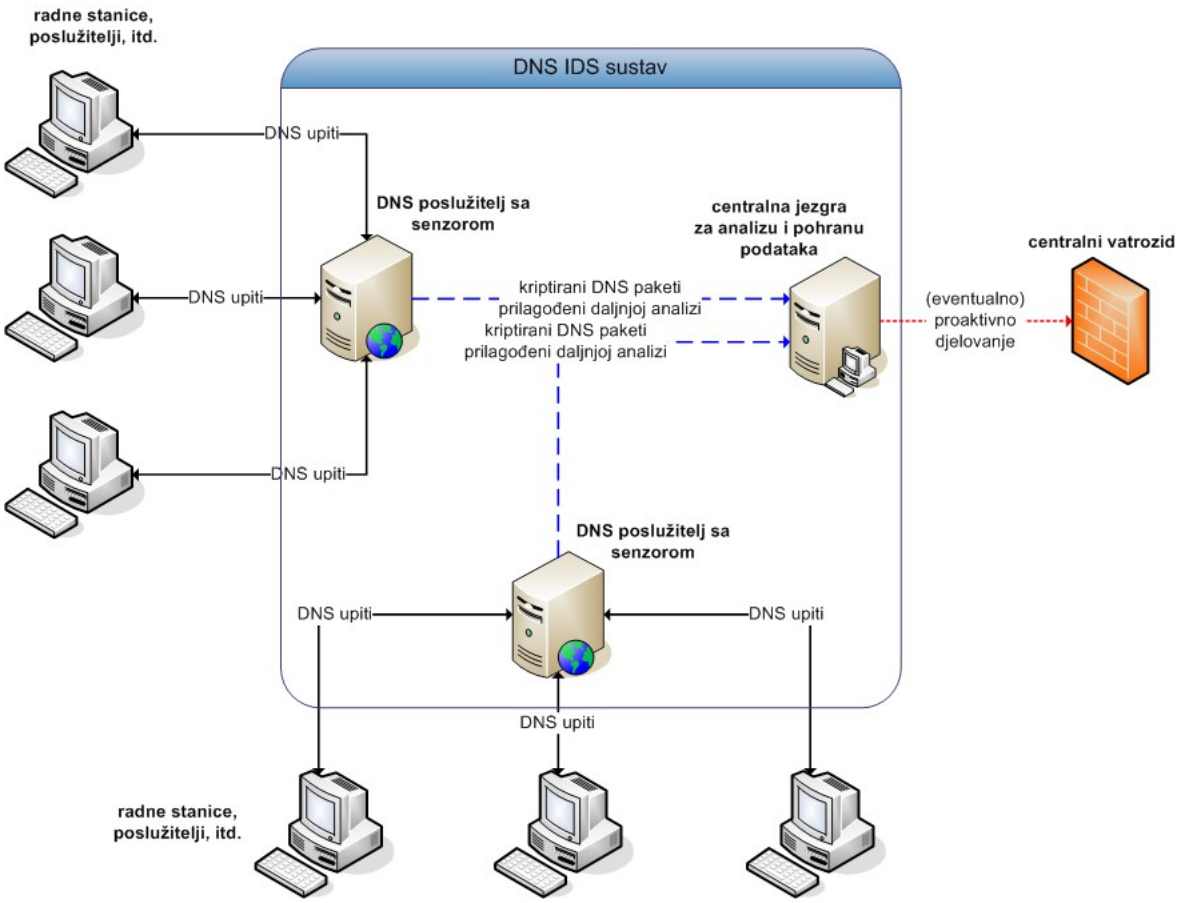
- centralna jezgra koja prihvaća kriptirane DNS pakete od senzora te obavlja sigurnosnu analizu istih, kao i pohranu podataka.

Mogu se definirati sljedeća ograničenja sustava:

- senzora mora biti jedan ili više,
- moguće je imati i senzor i poslužitelj u jednom (što je samostojeći način rada),
- u svakom pojedinom trenutku mora postojati samo jedan centralni poslužitelj.

Sa komunikacijskog gledišta (slika 3.2 prikazuje komunikacijski put unutar sustava), karakteristike sustava su sljedeće:

- originalni DNS upiti putuju neometano do svog cilja (DNS poslužitelja koji je autoritativan ili spremnički) i njihovo prisluškivanje je isključivo pasivno,
- prijenos enkriptiranih i dekodiranih DNS paketa prema centralnom poslužitelju se odvija jednosmjerno, paralelno (šalju se potencijalno istovremeno s različitih agenata) i asinhrono (s određenim neizbježnim kašnjenjem) naspram originalnih upita,
- centralna jezgra prispjele upite trenutno zaprima, no detaljno (sigurnosna analiza) ih obrađuje redom prispijeća, što je također u određenom kašnjenju naspram originalnog događaja,
- senzori nisu svjesni da li jezgra uopće radi i da li možda došlo do gubitaka paketa,
- jezgra ne zna o kojem je broju senzora riječ, već pasivno iščekuje.

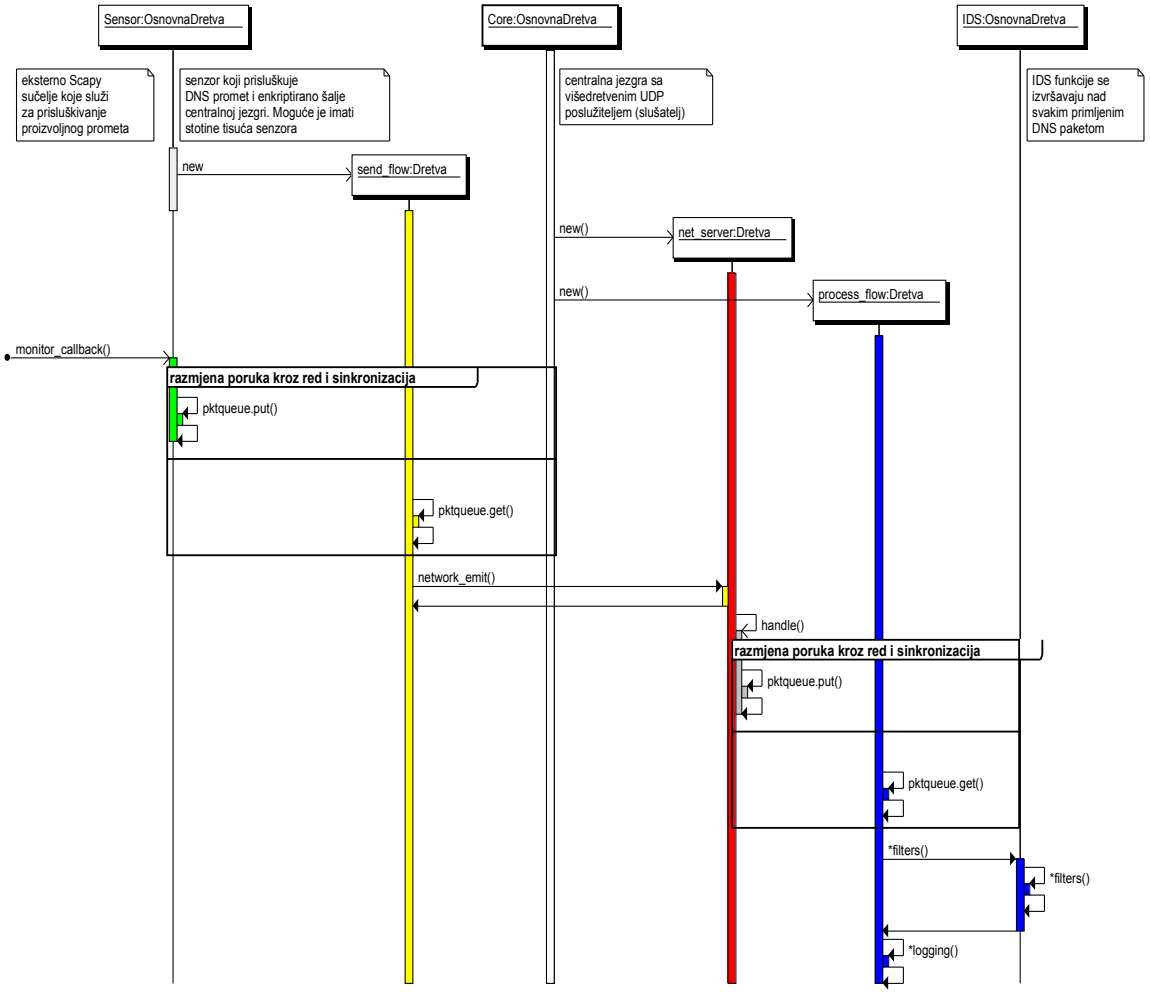


Slika 3.2: Arhitekturalni prikaz komunikacije u sustavu

Vremenskim slijedom dešavaju se sljedeće akcije u sustavu nakon prihvata jednog DNS paketa od strane jednog senzora (slika 3.3):

- unutar senzora:
  - pri prispjeću odnosno identificiranom DNS paketu unutar Scapy biblioteke, okida se callback procedura unutar agenta,
  - agent prispjeli DNS paket dekodira (pretvara iz sirovog oblika u odgovarajući prezentacijski oblik) te sprema u red poruka, nakon čega se callback procedura završava,
  - čim je red poruka neprazan (barem jedna poruka), zasebna dretva DNS agenta se budi (za razliku od callback procedure koja se poziva po potrebi, ona je stalno prisutna ali je u stanju čekanja dokle god nema paketa u redu poruka), prihvaća paket iz reda (čime on nestaje iz reda poruka) te ga serijalizira, enkriptira, računa zaštitnu sumu i odašilje prema centralnom poslužitelju,
- unutar centralnog poslužitelja:

- za svaki prispjeli paket se stvara nova dretva koja prihvaća paket, provjerava, dekriptira i sprema u red poruka, nakon čega dretva završava s radom,
- zasebna dretva se budi (stalno prisutna i u stanju čekanja dokle god nema paketa u redu poruka), daljnje obrađuje paket obavljajući analizu sigurnosnim filtarima i bilježeći eventualne uočene sigurnosne probleme.



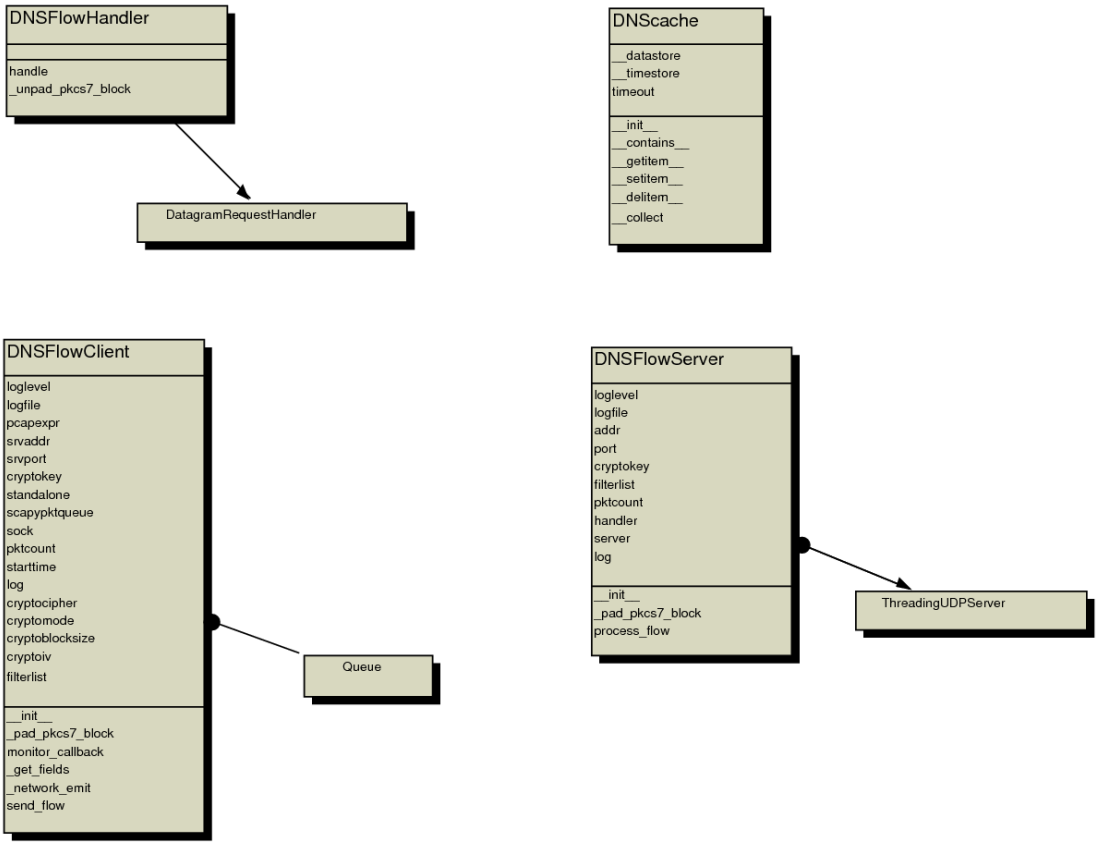
Slika 3.3: Tijek akcija obrade DNS paketa

Slika 3.4 donosi UML dijagram klasa sljedećih programskih komponenti:

- datoteka sniff_sensor.py:
  - klasa DNSFlowClient - sučelje prema Scapy biblioteci, pretvorba DNS paketa iz sirovog u prezentacijski oblik, serijalizacija te kriptiranje i slanje paketa,
- datoteka sniff_core.py - centralna jezgra s višedretvenim UDP poslužiteljem, dekriptiranjem, deserijalizacijom i sigurnosnim provjerama:

- klasa DNSFlowServer - nadležna klasa (inicijalna konfiguracija, postavljanje kriptografskih parametara, itd.),
- klasa DNSFlowHandler - centralna rutina za višedretveni UDP poslužitelj,

- datoteka sniff_dnscache.py:
  - klasa DNScache - jednostavni DNS međuspremnik koji se koristi za korelaciju DNS upita i odgovora,
  - datoteka sniff_filters.py - implementacija različitih pomoćnih metoda za vađenje i provjeru DNS podatka, kao i svih metoda za sigurnosnu analizu; koristi se u centralnom poslužitelju ili samostojećem senzoru.



Slika 3.4: Međuodnos programskih klasa

3.3. Otkrivanje problematičnog prometa

Funcionalno najsloženiji aspekt ove implementacije je nedvojbeno identifikacija sigurnosnih problema u uhvaćenim DNS paketima. Za pojedine metode je dovoljno promatrati izolirane pakete (npr. uočiti određene pogreške u upitu ili odgovoru), dok je za druge potrebno promatrati i sam kontekst, odnosno korelirati prethodne upite i odgovore na

njih. Same sigurnosne provjere se nužno moraju oslanjati na brojne metode nižeg stupnja koje dohvaćaju različite dijelove DNS paketa obavljajući pri tome sve potrebne provjere (postoje li uopće odjeljci i zastavice, da li je došlo do sistemske pogreške, da li je zapis u obliku teksta umjesto cjelobrojni, itd.).

Otkrivanje nepravilnosti je ostvareno kroz individualne filtre (programski kod) koji prepoznaju i izoliraju pojedine probleme, bilježeći sve informacije o paketu koji je uzrokovao pogrešku. Implementirane su sljedeće metode odnosno filtri:

- nepoznati TLD-ovi u QNAME zapisima: IANA definira točan popis vršnih domena koje se postoje; u slučaju da upit sadrži nepoznatu vršnu domenu, riječ je o pogrešci na strani klijenta najčešće uzrokovanoj krivom DNS konfiguracijom ili pak pogreškama u aplikaciji koja obavlja DNS upite,
- A-za-A upiti: riječ je o nepotrebnim upitima za razrješenjem IP adrese u IP adresu za koje je unaprijed očito da su suvišni, a takvi upiti predstavljaju ozbiljnu pogrešku u DNS aplikaciji,
- RFC1918 upiti: kada klijenti imaju privatne adrese, uobičajeno je da se upiti za privatnim adresama šalju prema lokalnom DNS poslužitelju, no u slučaju da takvi upiti "dalje" cure prema višim DNS poslužiteljima riječ je o ozbiljnoj pogrešci u DNS konfiguraciji jer nema potrebe da se oni prosljeđuju, a samo pridonose opterećenju svjetskih DNS poslužitelja,
- upiti s neispravnim DNS oznakama: DNS standard striktno propisuje da DNS oznake smiju biti (- označava raspon, a navodnici pojedinu grupu znakova): "a-z", "A-Z", "0-9" te znakovi "-_*./@"; u slučaju da upit sadrži neke netipične znakova poput dvotočke, točka-zareza i sl, riječ je o pogrešci,
- upiti za starim i eksperimentalnim zapisima: RFC1035 definira koji su DNS RR tipovi zastarjeli odnosno koji imaju eksperimentalno značenje, takvi se upiti u standardnoj komunikaciji ne bi smjeli pojavljivati,
- pokušaji prepisivanja spremnika: Microsoft DNS klijenti i poslužitelji su imali poznatu ranjivost MS06-041 koja je udaljenom napadaču omogućavala prepisivanje spremnika i izvršavanje proizvoljnog koda na pogođenim računalima; stoga je ovakve napade potrebno pravovremeno prepoznati i pravovremeno proaktivno djelovati,
- nepoznati OPCODE odnosno nepoznati tip upita: u slučaju da oznaka DNS operacije koja se traži nije QUERY, IQUERY, STATUS ili UPDATE, riječ je o tipu upita koji se ne bi smio desiti,
- greška u obliku upita: poslužitelj šalje odgovarajuću poruku o pogrešci kad upit ima pogrešku u formatu; ovaj filtar prepoznaje takve povratne poruke i bilježi povratnu informaciju koja sadrži i originalan upit,

- više od 3 odgovora s istim ID-om: postoji niz poznatih i efikasnih tehnika trovanja DNS međuspremnika lažiranim DNS odgovorima; u slučaju da se dogodi više od 3 odgovora s istim jedinstvenim identifikatorom, filtar se aktivira budući da je riječ o potencijalnom sigurnosnom napadu gdje se variranjem odgovora s istim ID-jem pokušava zatrovati udaljeni DNS poslužitelj,
- odgovor na nepoznati upit: filtar identificira odgovore koji nemaju odgovarajući DNS upit odnosno upite na upit koji nije nikad postavljen, budući da su takvi odgovori najčešće pokušaju trovanja DNS međuspremnika,
- odgovor na lažni upit: filtar provjerava nadolazeće odgovore i provjera da li kopija upita u tim odgovorima odgovara zaista postavljenim upitima ili je riječ o lažiranim odgovorima.

Za potrebe korelacije upita i odgovora napravljen je jednostavni DNS međuspremnik koji pamti upite određeno vrijeme; samo trajanje je podesivo, a tipično se koristi 300 sekundi za vrijeme života upamćenih upita. Ključ za razlikovanje pojedinih upita jest jedinstveni identifikator pojedinog upita (odnosno ID), odredišna adresa prema kojoj je odaslan upit i odredišni port. Međuspremnik omogućava i bilježenje odgovora za pojedini upit, odnosno brojanje odgovora.

3.4. Daljnji rad

U samom radu nisu implementirane neke funkcionalnosti za koje se smatralo da su od sekundarne važnosti i da nisu predmet diplomskog rada:

- anonimiziranje podataka: sustav bilježi niz podataka koji mogu ugroziti privatnost pojedinaca ako se zapisnici objave javno, odnosno koriste i izvan ustanove gdje su zabilježeni,
- različiti ključevi za različite klijente: trenutno svi senzori koriste isti dijeljeni ključ, što uzrokuje da centralni poslužitelj ne razlikuje pojedine senzore; također u slučaju krađe ključa s jednog senzora, cijeli je sustav sigurnosno ugrožen,
- različite metode enkripcije: sustav koristi isključivo AES-128-CBC metodu enkripcije podataka, dok bi poželjno bilo omogućiti mijenjanje veličine bloka, enkripcijski algoritam kao i ostalih parametara komunikacije,
- identifikacija rada centralnog poslužitelja: senzori nemaju načina ustanoviti da li centralni poslužitelj uopće radi budući da je komunikacija isključivo jednosmjerna,
- bilježenje sirovih paketa u PCAP obliku: iako samo Scapy sučelje omogućava bilježenje sirovih (nedekodiranih) DNS paketa, sam sustav bilježi isključivo u prezentacijskom obliku.

4. Rezultati i razmatranje

Prethodno opisani sustav distribuiranog prikupljanja i analize DNS prometa ostvaren je u programskom jeziku Python (senzor, centralni poslužitelj, komponenta za sigurnosnu analizu i DNS međuspremnik) te su prije puštanja u produkciju provedena odgovarajuća formalna testiranja sukladnosti DNS standardima kao i mjerenje utjecaja na vršno opterećenje DNS poslužitelja. U nastavku će se obratiti pažnja na korištene programe i metodologiju testiranja, kao i dobivene rezultate u praksi.

4.1. Formalno testiranje sustava

Za potrebe testiranja sukladnosti DNS standardima korišten je PROTOS *Security Testing of Protocol Implementations* (URL: http://www.ee.oulu.fi/research/ouspg/protos/) sustav koji omogućava testiranje rada DNS poslužitelja. Konkretno sustav provjerava reakciju DNS poslužitelja na skup od 10460 različitih upita, DNS klijenata na skup od 11138 različitih odgovora, kao i reakciju DNS poslužitelja na prijenos zone s 11022 varijacija. Ti testovi pokrivaju sve tipične situacije komunikacije DNS klijenata i poslužitelja, kao i međusobne komunikacije dvaju DNS poslužitelja. Sustav koji uspješno prolazi ove testove zadovoljava osnovni skup DNS standarda (RFC1035, RFC2929, RFC2136, RFC2671, RFC3007, RFC2845, RFC2065, RFC2874, RFC2535 te RFC2931).

Tijekom testiranja je korišten Bind 9 poslužitelj u inačici 9.6.0 s odgovarajućom konfiguracijom gdje je onemogućeno kontaktiranje ostalih DNS poslužitelja kako bi se minimizirale latencije i uklonila nepotrebna dodatna komunikacija nevezana uz osnovno testiranje. Relevantan blok konfiguracije je prikazan u tablici 4.1.

Tablica 4.1: Referentna konfiguracija Bind poslužitelja

```
options {
        directory "...";
        transfer-format many-answers;
        check-sibling no;
        recursion no;
        fetch-glue no;
        allow-recursion { none; };
        max-acache-size 128M;
};
```

Testiranje sukladnosti se sastojalo od 3 dijela:

- korištenja PROTOS sustava za stvaranje DNS upita (PROTOS je obavljao ulogu DNS klijenta) prema Bind 9 poslužitelju dok je sustav za nadzor i analizu bilježio i analizirao DNS promet,
- korištenja dig naredbe za slanje DNS upita prema PROTOS sustavu (PROTOS je obavljao ulogu DNS poslužitelja) dok je sustav za nadzor i analizu bilježio i analizirao DNS promet,
- korištenja dig naredbe za prijenos zone od PROTOS sustava (PROTOS je obavljao ulogu DNS poslužitelja) dok je sustav za nadzor i analizu bilježio i analizirao DNS promet.

Sustav za nadzor je uspješno obavio sve zadaće i ustanovio sve očekivane probleme.

U drugom dijelu testiranja, mjereno je vršno opterećenje DNS poslužitelja građenog oko Bind 9 poslužitelja[9] u situaciji sa i bez aktivnog sustava za nadzor. Stvorena je baza od 20 milijuna DNS upita sa sljedećom raspodjelom:

- 33% upita s konkretnim sadržajem iz lokalne zone (isključivo upiti prema A i CNAME zapisima),
- 33% upita za zapisima koji ne postoje u zoni ali imaju ispravnu lokalnu domenu,
- 33% upita koji ne postoje i uzrokuju povratnu pogrešku.

Korišten je Bind 9 poslužitelj u već opisanoj konfiguraciji (tablica 4.1), te alat queryperf koji je dio standardne Bind 9 distribucije (URL: https://www.isc.org/software/bind). Promatrana su tri osnovna slučaja:

- DNS poslužitelj bez dodatnog sustava za nadzor i analizu,
- DNS poslužitelj sa sustavom za nadzor koji bilježi samo incidente,
- DNS poslužitelj sa sustavom za nadzor koji bilježi sve DNS pakete sa svim detaljima kao i incidente.

Dobiveni rezultati (tablica 4.2) pokazuju vršne performanse DNS poslužitelja kroz qps (eng. *Queries per second*) vrijednost, odnosno broj uspješno odrađenih upita u sekundi. Tipične vrijednosti za standardni opterećeni DNS poslužitelj se kreću oko 2000 qps, dok vršni DNS poslužitelji primaju promet tipično između 5000 i 12000 qps [16]; shodno tome je i daleko manje primjetan utjecaj na ukupno opterećenje nego ovdje prikazani najgori slučaj. U tipičnoj upotrebi DNS poslužitelja su rijetke situacije da se zaprimi i odradi 20 milijuna DNS paketa unutar 15ak minuta; tijekom kasnijeg praktičnog testiranja je na

9 Korišteno je tipično PC računalo građeno oko Intel Core2Duo E8500 procesora na 3.16GHz, 4GB DDR2 CAS4 RAM na 800MHz te RAID1 građen od 2x SATA2 diskova Samsung Spinpoint F1 pojedinačnog kapaciteta 1TB.

centralnom FSB DNS poslužitelju zaprimljeno 40 milijuna paketa unutar čak 10 radnih dana.

Tablica 4.2: Utjecaj nadzora na DNS performanse

	DNS poslužitelj bez nadzora	**DNS poslužitelj s nadzorom, bilježe se samo incidenti**	**DNS poslužitelj s nadzorom, bilježe se svi DNS paketi i incidenti**
obrađenih upita u sekundi (qps)	26716.61	25089.52	25113.36
ukupno trajanje testiranja (sec)	748.60	797.15	796.39

Iz rezultata je vidljivo da je utjecaj nadzora na performanse DNS poslužitelja zanemariv, čak i u situacijama gdje se bilježe svi detalji o DNS prometu. U praksi se može očekivati da intenzivno pisanje po datotečnom sustavu ima nuspojavu povišenog vršnog opterećenja poslužitelja, no taj problem rješava instalacija čistih senzora bez IDS komponente.

4.2. Mjerenja u produkciji i diskusija rezultata

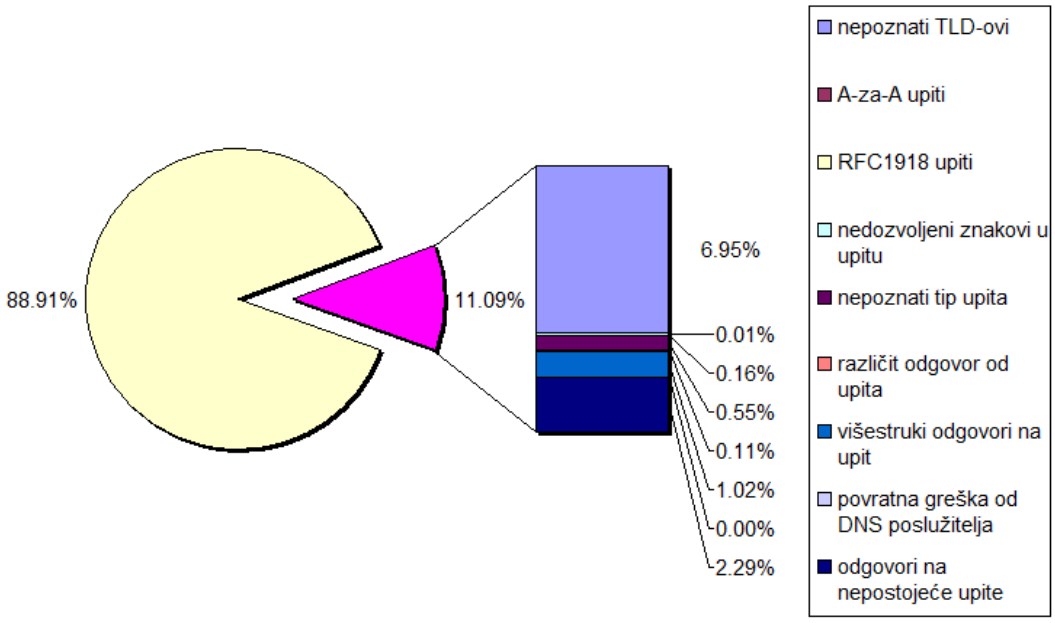
Za potrebe testiranja u stvarnom okruženju instaliran je na hobbit.fsb.hr, centralni DNS poslužitelj Fakulteta strojarstva i brodogradnje u Zagrebu (u daljnjem tekstu FSB). U razdoblju od 243 radna sata, pregledano je 39 milijuna dolaznih i odlaznih DNS paketa i zabilježeno preko 4 milijuna potencijalnih sigurnosnih prijetnji te 7 tisuća pogrešaka u DNS komunikaciji (uzrokovanih pogreškama u DNS aplikacijama udaljenih klijenata odnosno udaljenih poslužitelja). Sustav je tijekom rada stvorio datoteku veličine 4.1GB s kompletnim pregledom pojedinih sigurnosnih incidenata te statistikama o vlastitom radu (broj zaprimljenih paketa, sati rada, ukupni broj incidenata).

Skraćeni izvadak najzanimljivijih podataka, izvješća o radu sustava i uočenih problema u DNS prometu obuhvaća:

- 39044215 analiziranih dolaznih i odlaznih DNS paketa,
- 7115 kritičnih pogrešaka u DNS komunikaciji (kritične pogreške u formatu DNS paketa, pogreške u kompresiji DNS oznaka, itd.),
- 4217793 potencijalnih sigurnosnih prijetnji, što je 10.80% od ukupnog prometa s prosječno 17357 zabilježenih incidenata po radnom satu; a od toga:
  - 293216 nepoznatih TLD zapisa,
  - 211 A-za-A upita (upiti za pronalaženjem IP adrese iz oblika koji već jest IP adresa),
  - 3749951 upita za privatnim RFC1918 adresama,

- 6624 upita s pogreškama (nedozvoljeni znakovi) u DNS oznakama,
- 0 upita za starim RR zapisima,
- 0 upita za eksperimentalnim RR zapisima,
- 0 pokušaja iskorištavanja MS06-041 ranjivosti,
- 23352 upita s nepoznatim tipom upita (pretežno pokušaji dinamičkih DNS upita),
- 127 identificiranih pogrešaka u obliku paketa (povratna identifikacija od strane DNS poslužitelja),
- 43013 višestrukih odgovora na pojedini DNS upit (mogući pokušaji trovanja DNS međuspremnika),
- 4667 odgovora koji ne sadrže identični upit onome koji je poslan (lažirani odgovori, odnosno odgovori gdje je polje upita pogreškom neispravno popunjeno),
- 96632 odgovora koji su ili zakašnjeli (preko 300 sekundi) ili su lažirani odgovori za upite koji nisu nikad poslani.

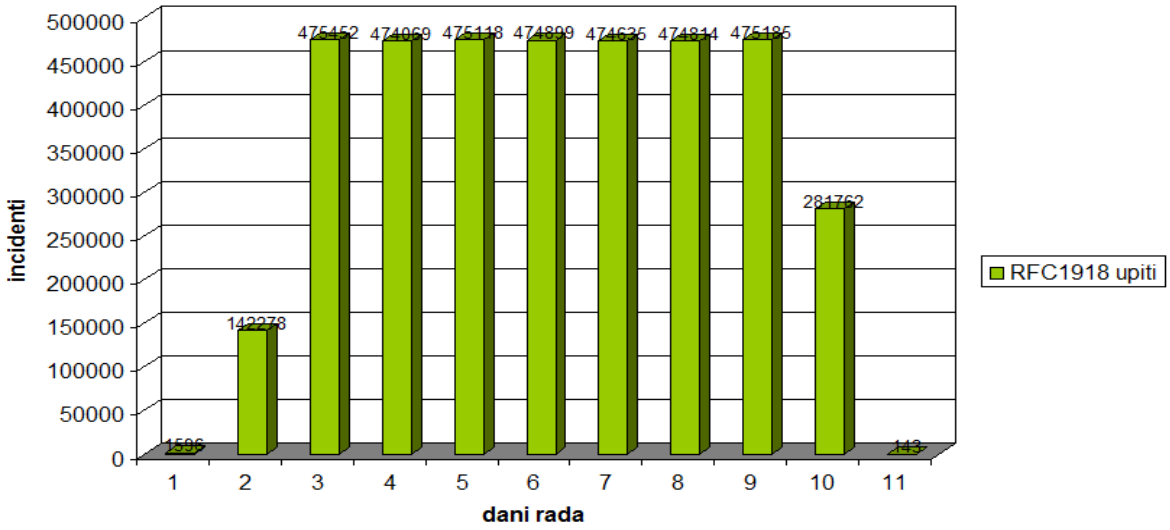
Iz slike 4.1 je vidljivo da promet koji se odnosi na razrješenje privatnih adresa zauzima 89% incidenata (utvrđeno je u prethodnim istraživanjima da 1.61% svjetskog DNS prometa predstavlja curenje RFC1918 upita prema vršnim DNS poslužiteljima [13]), no podatak je još zanimljiviji kad se uzme u obzir da je mreža FSB isključivo u javnom rasponu 161.53.116.0/22, odnosno da se privatne adrese standardno ne koriste. Može se zaključiti da je riječ o prometu koji je uzrokovan neispravnim konfiguracijama uređaja iza kojih se nalaze pojedine privatne mreže, gdje se DNS upiti ne razrješavaju lokalno već bivaju proslijeđeni na nadležne poslužitelje. Mjerenje je potvrdilo da je značajan broj incidenata vezan uz "curenje" privatnih adresa te da je potrebno posebnu pažnju posvetiti upravo ispravnom filtriranju takvih upita.



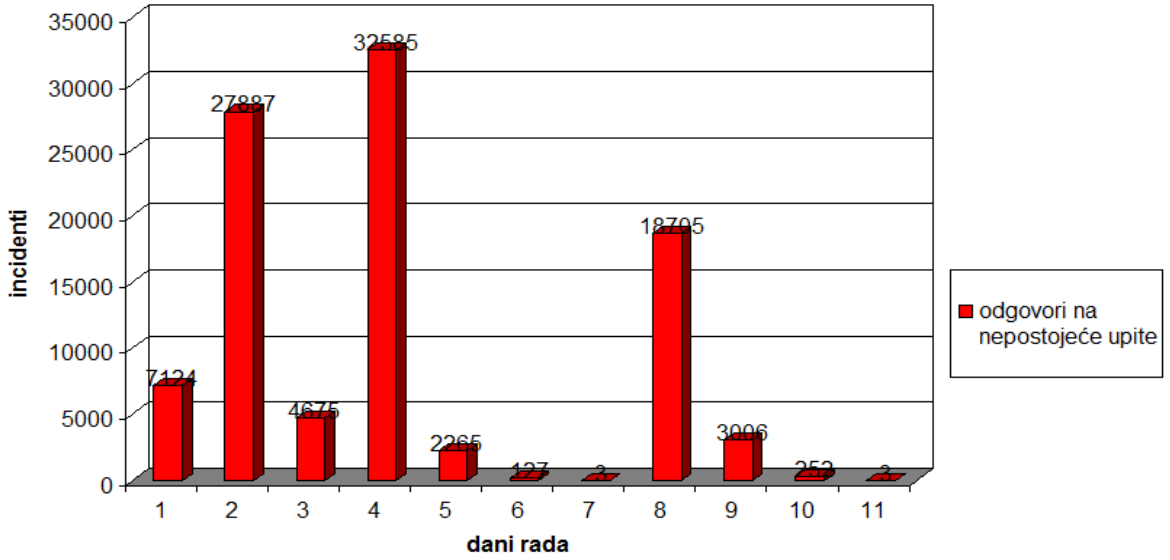
Slika 4.1: Raspodjela ukupnog broja incidenata na FSB-u

Iz mjerenja se također može zaključiti da je ukupni broj incidenata razmjerno nizak naspram ukupnog DNS prometa (svega 10% prometa), što je različito od rezultata s vršnih DNS poslužitelja: prema dosadašnjim mjerenjima na vršnim poslužiteljima, svega 2% je legitiman promet [14]. Razlog se može tražiti u činjenici da je mreža FSB strogo kontrolirana okolina (ažurne radne stanice, antivirusne zaštite, legalne aplikacije, itd.), pa je broj pogrešaka uzrokovanih neispravnim konfiguracijama znatno manji nego onaj na poslužiteljima koji su otvoreni prema cijelom svijetu.

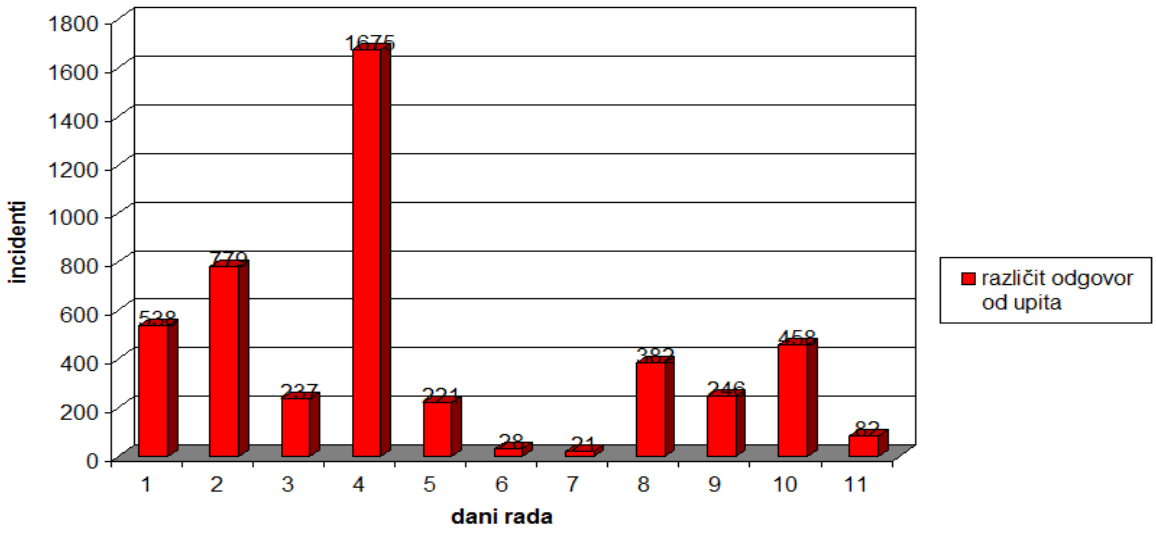
Pregled zapisnika ukazuje da je 3881778 incidenata uzrokovano iz lokalne mreže FSB, što je 92.11% incidenata. Kad se zanemare upiti za privatnim adresama (prikazani na slici 4.2), dobiva se da je 276254 (58.08%) preostalih sigurnosnih incidenata uzrokovano iz lokalne mreže, a 199421 izvana (41.92%). Bitno je primijetiti kako preostali incidenti imaju značajnije sigurnosne posljedice, iako se rjeđe pojavljuju (vidjeti sliku 4.3, 4.4 i 4.5): primjerice u slučaju da uspije pokušaj trovanja DNS poslužitelja, jedan DNS paket može zatrovati sve klijente koji koriste poslužitelj i tako uzrokovati dugotrajne i dalekosežne posljedice. Stoga iako RFC1918 upiti imaju tendenciju stalnog opterećivanja DNS poslužitelja, oni se jednostavno mogu filtrirati i riješiti, dok to za lažne upite, lažne odgovore i ostale pokušaje trovanja DNS poslužitelja nije nimalo jednostavno.



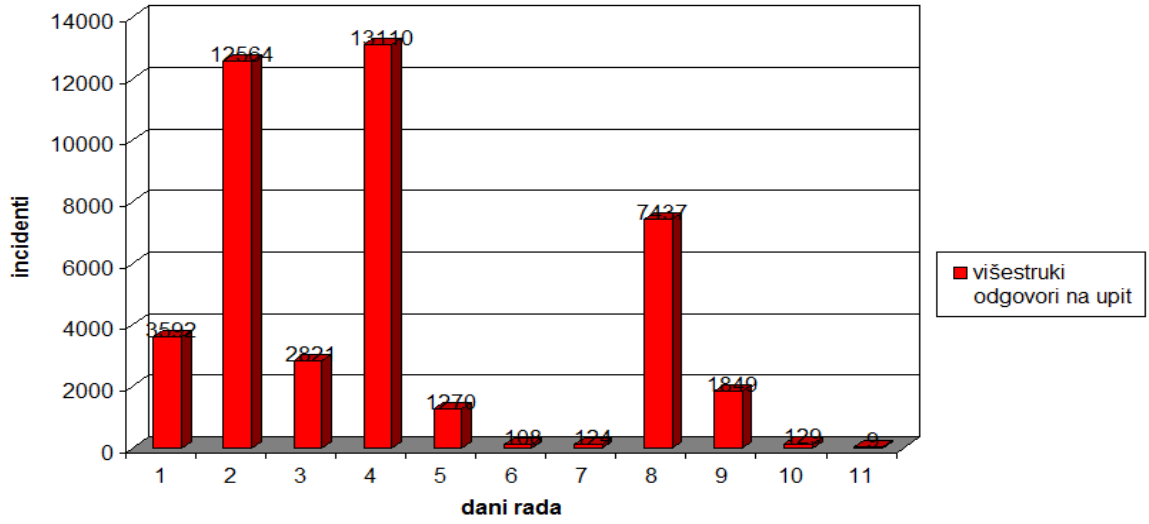
Slika 4.2: RFC1918 upiti (privatne adrese)



Slika 4.3: Odgovori na nepostojeće upite



Slika 4.4: Odgovori različiti od upita



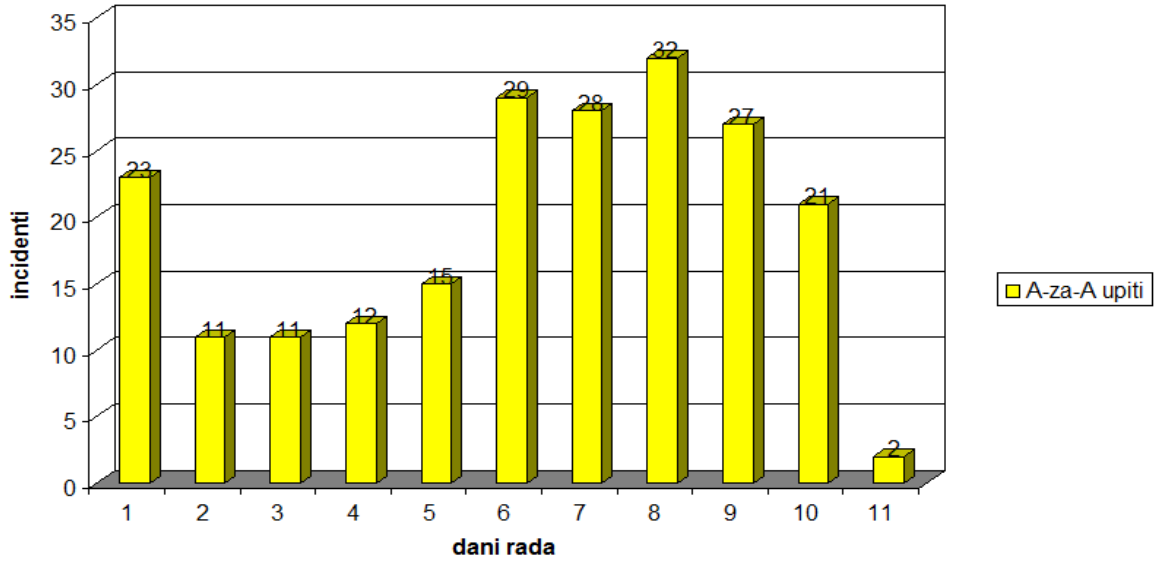
Slika 4.5: Višestruki odgovori na upit

Sigurnosni incidenti od sekundarne važnosti su:

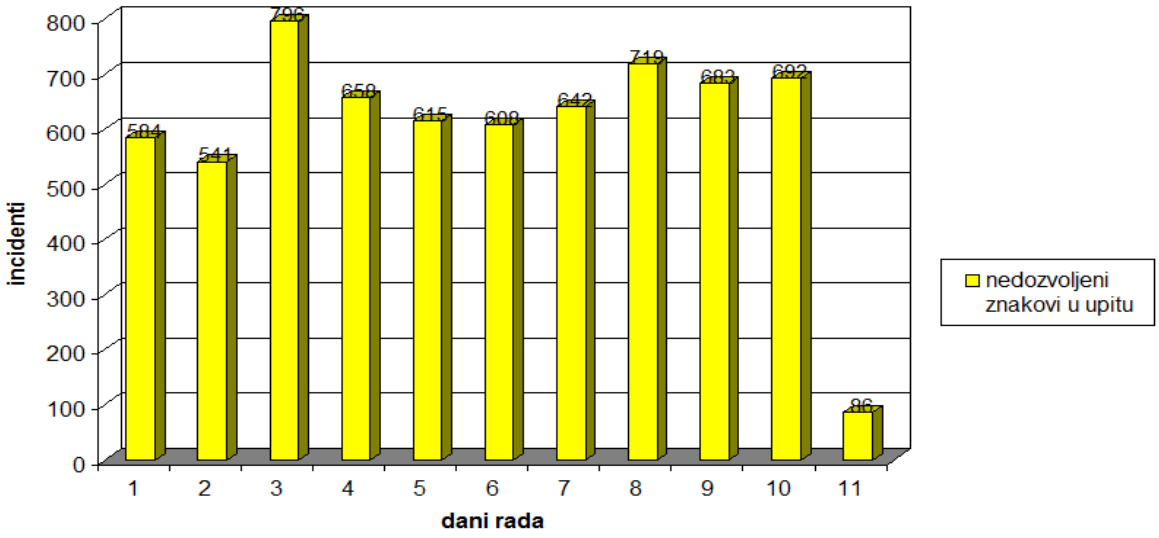
- A-za-A upiti (prikazani na slici 4.6),
- upiti s nedozvoljenim znakovima (slika 4.7),
- nepoznati tip upita (slika 4.8).

Prva dva tipa incidenata su isključivo vezani uz pogreške u aplikacijama koje koriste DNS usluge; jedan od uzročnika je primjerice Nod32 antivirusni program koji pokušava poslati

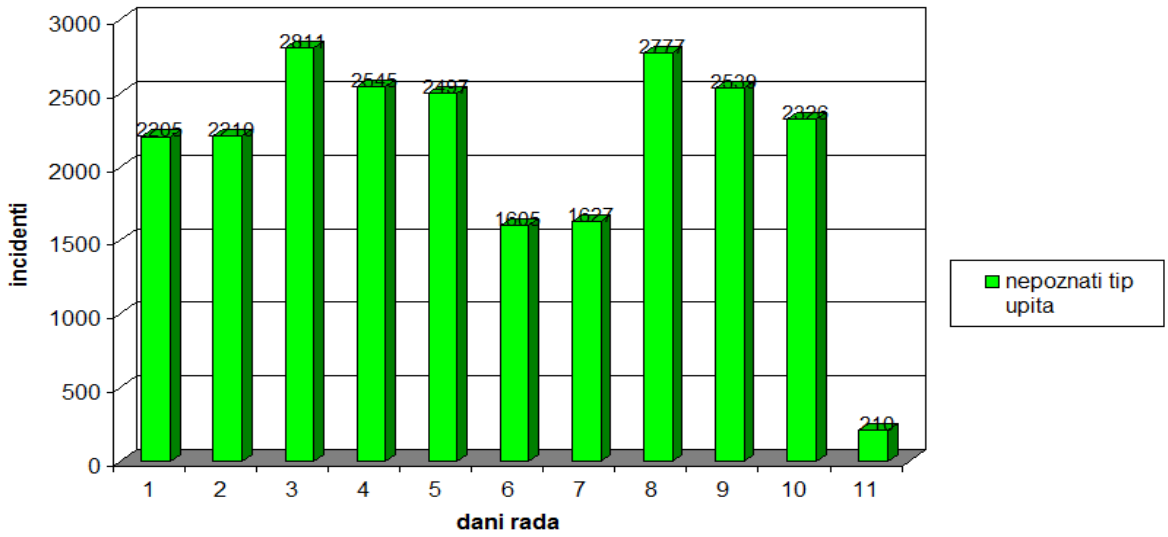
upit s portom u imenu poslužitelja s antivirusnim definicijama. Što se tiče upita s nepoznatim kodom operacije, riječ je o DNS klijentima (uglavnom Microsoft Windows računala) koji su neispravno konfigurirani i šalju dinamičke DNS upite iako se dinamički DNS ne koristi u mreži FSB. Stoga se te upite može promatrati kao vrstu DNS zagađenja koja opterećuje DNS poslužitelj bez drugih posljedica.



Slika 4.6: A-za-A sigurnosni incidenti

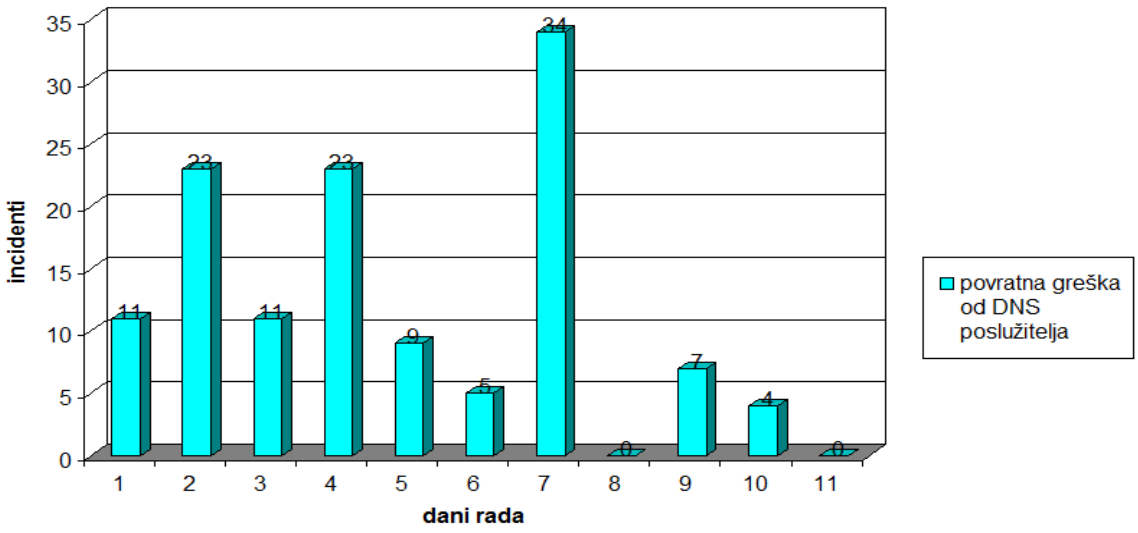


Slika 4.7: Nedozvoljeni znakovi u upitu



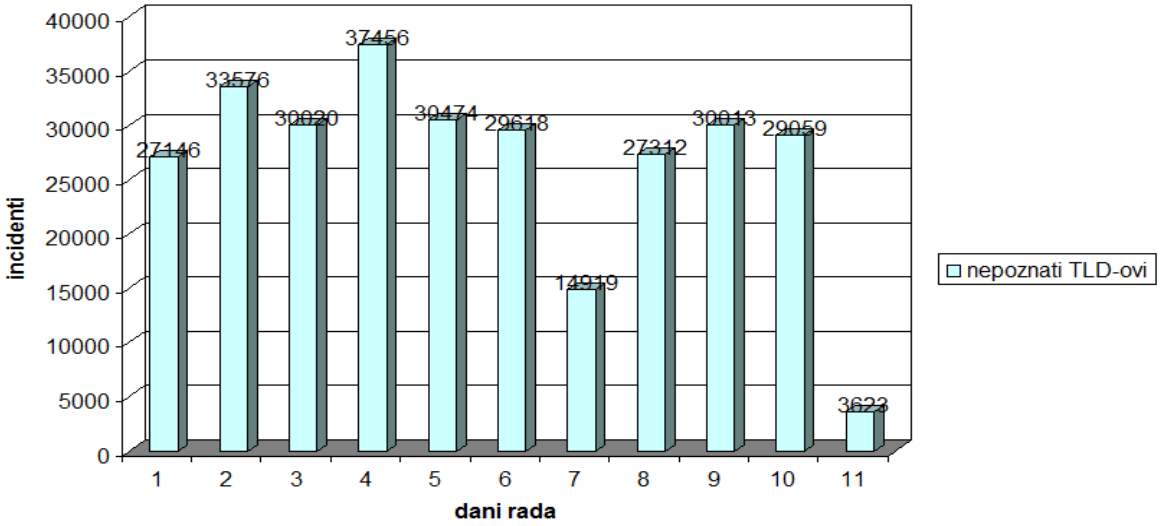
Slika 4.8: Nepoznati tip upita

Osim navedenih problema, sustav je otkrio i statistički manje važan broj pogrešaka u obradi DNS paketa koje je DNS poslužitelj prijavio (slika 4.9). Iako su to pogreške koje su neobične jer je riječ o pogreškama u obliku DNS upita (primjerice neispravna kompresija oznaka), ni one ne bi smjele imati sigurnosni značaj budući da svaki DNS poslužitelj mora obavljati temeljitu provjeru prispjelih DNS paketa. U slučaju da takva provjera nije ispravna, ovakav promet bi mogao uzrokovati neispravno funkcioniranje i eventualni prestanak rada DNS poslužitelja.



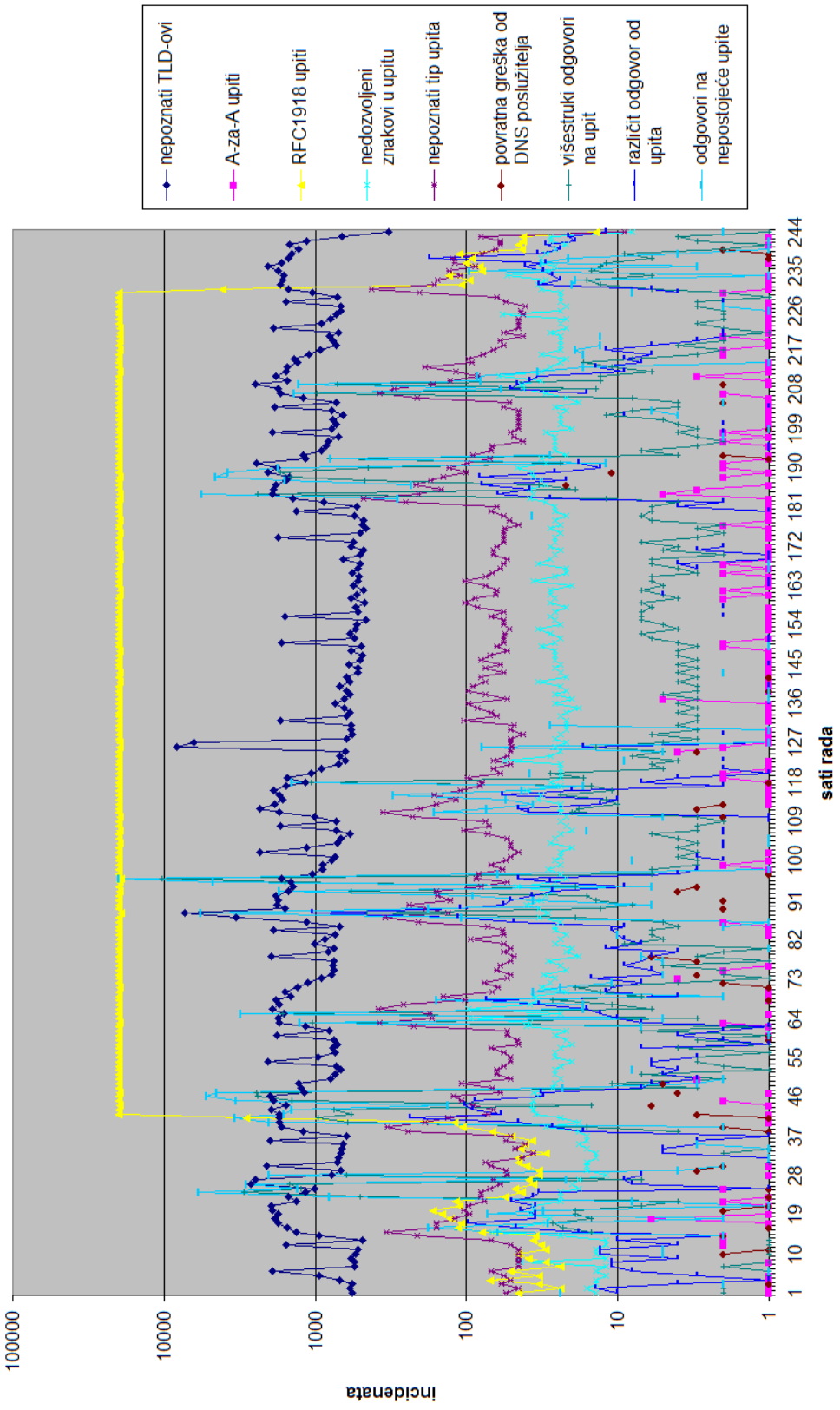
Slika 4.9: Povratna pogreška od DNS poslužitelja

Među najbrojnijim pogreškama uočenim u DNS prometu su nedvojbeno upiti s krivim zapisima za vršnu domenu (slika 4.10). Kao i s RFC1918 upitima, uzrok je neispravna konfiguracija računala ili poslužitelja koji šalju upite s neispravnim DNS oznakama koje nisu u FQDN obliku ili imaju neke dodatne oznake proizašle iz pogrešaka u lokalnoj DNS konfiguraciji. Takvi upiti također opterećuju centralni DNS poslužitelj, njihovo filtriranje jest teško (praktički nemoguće), te najčešće bivaju proslijeđeni dalje u svijet čime pridonose općem zagađenju DNS prometa. Navedene je pogreške potrebno pravovremeno uočiti i riješiti ispravljanjem podešenja pojedinih uzročnika. Upravo ovaj sustav omogućava lako lociranje tih uređaja.



Slika 4.10: Nepoznate vršne domene

Opći vremenski pregled (slika 4.11) pokazuje da se svi tipovi incidenata osim RFC1918 upita slijede vrlo slične vremenske uzorke kroz svaki pojedini dan, dok RFC1918 upiti bivaju slani redovno, u velikim količinama kroz znatno veći vremenski period. Korelacijom vremena i daljnjim istraživanjem uzroka ustanovljeno je kako početak slanja znatnog RFC1918 prometa odgovara početku radnog tjedna i paljenju svih računala na FSB, dok završetak perioda i pad količine rečenih upita odgovara kraju radnog tjedna.



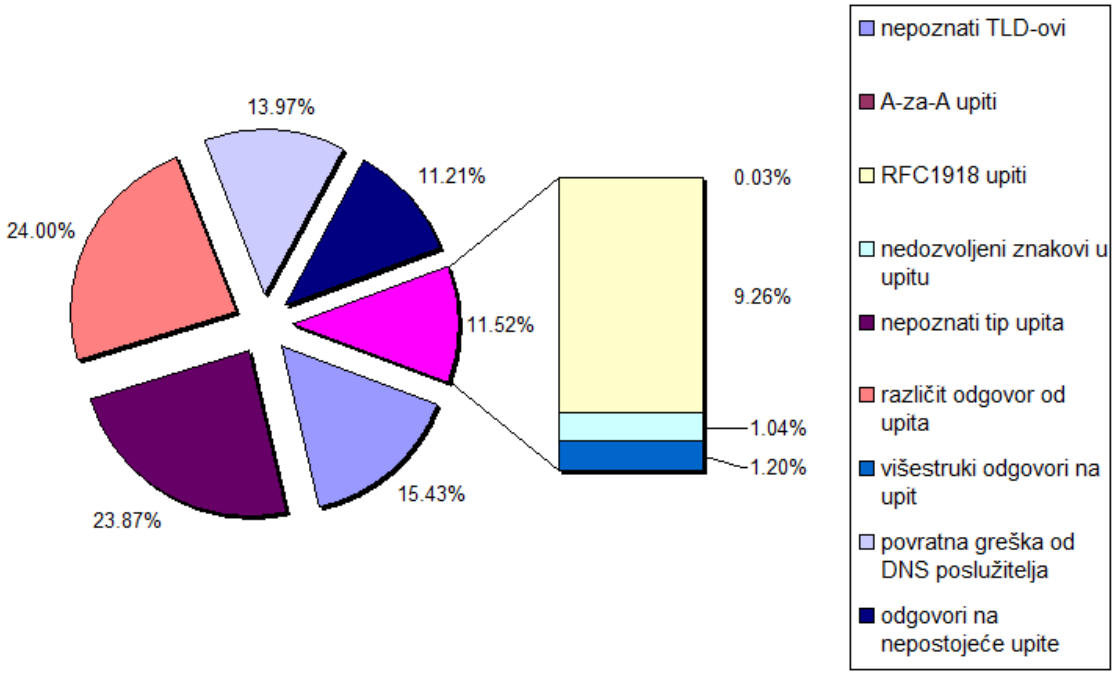
Slika 4.11: Skupni prikaz zabilježenih incidenata na FSB-u

Drugi dio mjerenja proveden je na računalu gandalf.zemris.fer.hr, centralnom DNS poslužitelju Zavoda za elektroniku, mikroelektroniku, računalne i inteligentne sustave Fakulteta elektrotehnike i računarstva u Zagrebu (u daljnjem tekstu ZEMRIS). ZEMRIS predstavlja manju radnu grupu s obzirom na veličinu okruženja od tek 60-ak aktivnih računala, pa je shodno tome moguće očekivati i drukčije obrasce zabilježenih nepravilnosti u DNS prometu.

U razdoblju od 229 radnih sati, pregledano je 12 milijuna dolaznih i odlaznih DNS paketa i zabilježeno 35 tisuća potencijalnih sigurnosnih prijetnji te 5 tisuća pogrešaka u DNS komunikaciji (uzrokovanih pogreškama u DNS aplikacijama udaljenih klijenata odnosno udaljenih poslužitelja). Skraćeni izvadak najzanimljivijih podataka, izvješća o radu sustava i uočenih problema u DNS prometu obuhvaća:

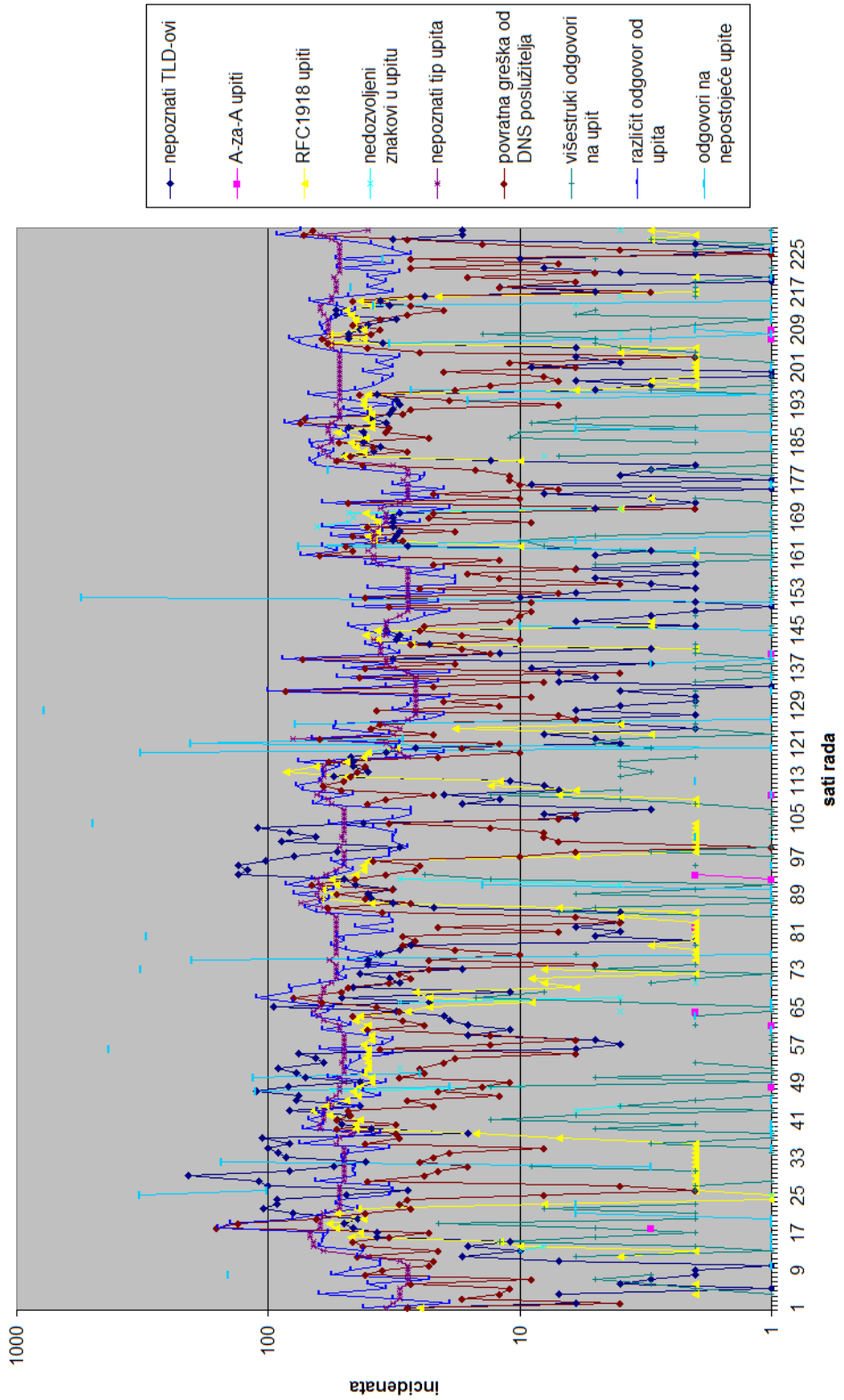
- 12950211 analiziranih dolaznih i odlaznih DNS paketa,
- 4794 kritičnih pogrešaka u DNS komunikaciji (kritične pogreške u formatu DNS paketa, pogreške u kompresiji DNS oznaka, itd.),
- 45971 potencijalnih sigurnosnih prijetnji, što je 0.35% od ukupnog prometa s prosječno 200 zabilježenih incidenata po radnom satu; a od toga:
  - 7093 nepoznatih TLD zapisa,
  - 16 A-za-A upita (upiti za pronalaženjem IP adrese iz oblika koji već jest IP adresa),
  - 4255 upita za privatnim RFC1918 adresama,
  - 476 upita s pogreškama (nedozvoljeni znakovi) u DNS oznakama,
  - 0 upita za starim RR zapisima,
  - 0 upita za eksperimentalnim RR zapisima,
  - 0 pokušaja iskorištavanja MS06-041 ranjivosti,
  - 10974 upita s nepoznatim tipom upita (pretežno pokušaji dinamičkih DNS upita),
  - 6420 identificiranih pogrešaka u obliku paketa (povratna identifikacija od strane DNS poslužitelja),
  - 551 višestrukih odgovora na pojedini DNS upit (mogući pokušaji trovanja DNS međuspremnika),
  - 11032 odgovora koji ne sadrže identični upit onome koji je poslan (lažirani odgovori, odnosno odgovori gdje je polje upita pogreškom neispravno popunjeno),
  - 5154 odgovora koji su ili zakašnjeli (preko 300 sekundi) ili su lažirani odgovori za upite koji nisu nikad poslani.

Za razliku od rezultata mjerenja na FSB-u (slika 4.1) gdje prevladava RFC1918 promet kao glavni tip problema, raspodjela prometa na ZEMRIS-u (slika 4.12) pokazuje da su skoro podjednako zastupljeni svi tipovi incidenata. 93.97% uočenih incidenata ima izvorište u lokalnoj mreži, no broj RFC1918 upita je praktički zanemariv što je uzrokovano minimalnim brojem privatnih podmreža na samom ZEMRIS-u. Veliki broj nepoznatih tipova upita ukazuje na dominantnost Microsoft Windows računala koja pokušavaju poslati dinamičke DNS upite na DNS poslužitelj iz SOA zapisa za zemris.fer.hr domenu, gandalf.zemris.fer.hr.



Slika 4.12: Raspodjela ukupnog broja incidenata na ZEMRIS-u

Izrazito mali broj incidenata (svega 0.35% od ukupnog DNS prometa) možemo objasniti uniformnom raspodjelom operacijskih sustava i njihovog načina korištenja, što je karakteristično za manje radne okoline. Isti trend ponašanja je vidljiv na slici 4.13, kao i detalj da su incidenti uglavnom ravnomjerno raspoređeni kroz pojedini radni dan. Što se tiče upita za nepoznatim TLD-ovima, naspram ostalih incidenata oni su snažno zastupljeni i vrlo se pravilno ponavljaju kroz dan: sastoje se od nepravilno zadanih SRV upita (računalo sauron.zemris.fer.hr) te upita koji ponajviše završavaju na "local" (greške u DNS konfiguraciji, primjerice na računalu adamanta.zemris.fer.hr), "wpad" (riječ je o pronalaženju HTTP/FTP posredničkog poslužitelja, odnosno eng. *Web Proxy Autodiscovery Protocol*) i "soc-8" (računalo soc-7.zemris.fer.hr, gdje zbog pogreške nije unesen DNS sufiks u konfiguraciji domene).



Slika 4.13: Skupni prikaz zabilježenih incidenata na ZEMRIS-u

5. Zaključak

Predmet ovog diplomskog rada je bila priprema, dizajn, izrada i praktično testiranje sustava distribuiranog prikupljanja i sigurnosne analize DNS prometa, budući da specijalizirani sustavi otkrivanja sigurnosnih prijetnji prema DNS poslužiteljima uopće ne postoje (kao što je i pokazano u pripremnom dijelu rada, poglavlju 3).

Tijekom praktičnog rada u potpunosti su ostvarene sljedeće komponente sustava u programskom jeziku Python:

- senzor koji omogućava pasivno prisluškivanje proizvoljnog unicast i multicast DNS prometa u TCP i UDP obliku, osnovnu obradu podataka i njihovo kriptirano slanje prema centralnom poslužitelju,
- centralni poslužitelj koji prima podatke od jednog ili više senzora, provjerava, dekriptira i daljnje obrađuje, lokalno spremajući rezultate,
- komponenta za detaljnu sigurnosnu analizu koja omogućava prepoznavanje 12 tipova sigurnosnih napada; moguće ju je koristiti u svakom senzoru (za samostojeći rad) ili u centralnom poslužitelju (obrada svih prispjelih informacija),
- jednostavni DNS međuspremnik koji omogućava kontekstualnu analizu upita i odgovora.

Također je osmišljen i razvijen vlastiti protokol za kriptiranu komunikaciju sa zaštitnim sumama koji udovoljava zahtjevima niskih latencija i mogućnosti djelotvornog prijenosa memorijskih struktura preko mreže. Sukladnost važećim DNS standardima provjerena je koristeći PROTOS alat (više u poglavlju 4.1), a utvrđeno je i da je utjecaj na performanse poslužitelja zanemariv u tipičnim opterećenjima karakterističnim za DNS poslužitelje.

Dobiveni rezultati nadzora potvrđuju do sada ustanovljene podatke o značajnom broju različitih nepravilnosti u DNS prometu (više o sigurnosnim prijetnjama je moguće pročitati u poglavlju 2.6). Tijekom testiranja sustav je sigurnosno analizirao više od 1000 DNS paketa u sekundi bez značajnog porasta opterećenja na centralnom poslužitelju i bez ikakvih negativnih utjecaja na njegov normalan rad. U razdoblju od 243 radna sata, pregledano je ukupno 39 milijuna dolaznih i odlaznih DNS paketa i zabilježeno preko 4 milijuna potencijalnih sigurnosnih prijetnji te 7 tisuća pogrešaka u DNS komunikaciji.

S obzirom na odlične rezultate, stabilan rad i znatnu količinu prikupljenih podataka (nalaze se na priloženom mediju, te su obrađeni u poglavlju 4), pokazalo se da izrađeni sustav može učinkovito prepoznati različite mrežne i sigurnosne probleme koji se manifestiraju na lokalnim DNS poslužiteljima. Takve nepravilnosti u prometu su uzrokovane neispravnim mrežnim konfiguracijama i različitim pogreškama u aplikacijama, a otkrivaju se vrlo teško zbog nedostatka specijaliziranih alata. Dodatni je problem što se takav DNS

promet najčešće širi prema nadležnim svjetskim DNS poslužiteljima gdje uzrokuje nepotrebna opterećenja, stoga ga je bitno rano otkriti i ukloniti njegove uzroke.

Na osnovi rezultata ustanovljen je i nemali broj ozbiljnih sigurnosnih incidenata uzrokovanih namjernim napadima iz stranih mreža. DNS poslužitelji tipično prepoznaju i odbacuju tek dio ovih pokušaja, što je i potvrđeno tijekom pripremnog dijela ovog rada. Takvi DNS paketi u slučaju uspješnog napada mogu uzrokovati dugotrajne i ozbiljne posljedice trovanja lokalnog DNS poslužitelja i njegovih klijenata, preusmjerujući korisnike u skladu s napadačevim željama. Također je otkriveno da velik broj uobičajenih korisničkih aplikacija poput antivirusnih alata i Web preglednika uzrokuje brojne neispravne DNS upite. To su primjerice upiti koji imaju nedozvoljene znakove u DNS oznaci, upiti za saznavanjem IP adrese iz IP adrese, upiti koji imaju nepostojeću vršnu domenu u oznaci, itd.

Naposljetku pokazalo se da je moguće napraviti mrežni sustav dobrih performansi u jeziku visoke razine poput Pythona, koristeći napredne tehnike poput callbackova, višedretvenog rada i redova poruka. Python se pokazao i kao iznimno kvalitetna platforma koja je omogućila stvaranje potpuno otvorenog i nadogradivog sustava.

S obzirom na brojnost i kritičnost spomenutih DNS problema (uzrokovanih lokalno i izvana), budući rad na ovakvom sustavu bi uključivao razvoj proaktivne komponente koja bi s obzirom na kritičnost pojedinih incidenata omogućavala lokalno ili globalno blokiranje uzročnika, kao i eventualne daljnje akcije.

6. Literatura

1. P. Mockapetris: *RFC1034: Domain names - concepts and facilities*, URL: http://www.ietf.org/rfc/rfc1034.txt (11/1987)
2. J. Postel: *RFC1591: Domain Name System Structure and Delegation*, URL: http://www.ietf.org/rfc/rfc1591.txt (3/1994)
3. P. Mockapetris: *RFC1035: Domain names - implementation and specification*, URL: http://www.ietf.org/rfc/rfc1035.txt (11/1987)
4. P. Vixie: *RFC2671: Extension Mechanisms for DNS (EDNS0)*, URL: http://www.ietf.org/rfc/rfc2671.txt (8/1999)
5. IETF: *RFC1123: Requirements for Internet Hosts -- Application and Support*, URL: http://www.ietf.org/rfc/rfc1123.txt (10/1989)
6. Steve Gibbard: *Geographic Implications of DNS Infrastructure Distribution*, URL: http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_10-1/101_dns-infrastructure.html (2005)
7. R. Elz, R. Bush: *RFC2181: Clarifications to the DNS Specification*, URL: http://www.ietf.org/rfc/rfc2181.txt (7/1997)
8. C. Everhart, L. Mamakos, R. Ullmann, P. Mockapetris: *RFC 1183: New DNS RR Definitions*, URL: http://www.ietf.org/rfc/rfc1183.txt (10/1990)
9. P. Mockapetris: *RFC1101: DNS Encoding of Network Names and Other Types*, URL: http://www.ietf.org/rfc/rfc1101.txt (4/1989)
10. D. Eastlake 3rd, E. Brunner-Williams, B. Manning: *RFC2929: Domain Name System (DNS) IANA Considerations*, URL: http://www.ietf.org/rfc/rfc2929.txt (11/2000)
11. Daniel Julius Bernstein: *DNS forgery*, URL: http://cr.yp.to/djbdns/forgery.html (2004)
12. Amit Klein: *BIND 9 DNS Cache Poisoning*, Trusteer, URL: http://www.trusteer.com/files/BIND_9_DNS_Cache_Poisoning.pdf (6/2007)
13. Duane Wessels: *Is Your Caching Resolver Polluting the Internet?*, CAIDA & The Measurement Factory, Inc., URL: http://dns.measurement-factory.com/writings/wessels-netts2004-paper.pdf (2004)
14. Duane Wessels, Marina Fomenkov: *Wow, That's a Lot of Packets*, CAIDA & The Measurement Factory, Inc., URL: http://dns.measurement-factory.com/writings/wessels-pam2003-paper.pdf (2003)
15. Dinko Korunić: *DNS priručnik*, URL: http://dkorunic.net/pdf2/DNS-prirucnik.pdf (2007)

16. Nevil Brownlee, Kc Claffy, Evi Nemeth: *DNS Damage - Measurements at a Root Server*, URL: http://www.caida.org/publications/presentations/ietf0112/dns.damage.html (2001)

7. Dodatak A: Sadržaj priloženog medija (CD/DVD)

Na priloženom mediju pohranjeni su podaci korišteni pri izradi rada i svi postignuti rezultati, a logički su organizirani prema smislu (vidjeti tablicu 7.1).

Tablica 7.1: Sadržaj priloženog medija

R. br.	Direktorij/datoteka	Sadržaj
1.	PROCITAJ_ME.TXT	Informacije o sadržaju medija
2.	/dok	Tekst rada u izvornom formatu
3.	/dok/Korunic_Prikupljanje_i_analiza_DNS_prometa.doc	Tekst rada u Microsoft Word formatu
4.	/dok/Korunic_Prikupljanje_i_analiza_DNS_prometa.pdf	Tekst rada u PDF formatu s oznakama poglavlja
5.	/dok/Korunic_Prikupljanje_i_analiza_DNS_prometa.ps	Tekst rada u Postscript formatu
6.	/dok/Korunic_Prikupljanje_i_analiza_DNS_prometa.ppt	Prezentacija
7.	/dok/izvori	Sadržaj izvora upotrebljavanih u radu
8.	/izvorni_kod	Izvorni kod izrađenih programa s uputama o radnoj okolini, komentarima, prevodiocima i sl.
9.	/izvorni_kod/struktura.doc	Opis strukture podataka
10.	/programi	Izvršni programi, skripte
11.	/programi/windows	Programi za Windows OS
12.	/programi/linux	Programi za Linux OS
13.	/rezultati	Ulazni podaci i rezultati

8. Dodatak B: Upute za instalaciju

Na CD-u se u odgovarajućoj komprimiranoj arhivi (ZIP za Windows OS, odnosno TGZ za Linux/Unix OS) nalazi cjelokupna distribucija programa zajedno sa Scapy Python modulom koji sadrži specijalne izmjene, pa stoga nije moguće koristiti eventualnu sistemsku inačicu.

Za ispravan rad senzora (sniff_sensor.py) i centralne jezgre (sniff_core.py) nužno je predinstalirati sljedeće programe:

- Pyton 2.5 interpreter sa svim standardnim modulima (moguće je koristiti i bilo koji noviji osim Python 3.0),
- PyCrypto (The Python Cryptography Toolkit) - kriptografski Python modul,
- IPy - Python modul za baratanje s IP adresama i rasponima.

Za uspješno postavljanje sustava potrebno je sljedeće:

- napraviti radni direktorij u kojem će se nalaziti programi, moduli i sistemski zapisnici,
- pozicionirati se u taj direktorij,
- otpakirati odgovarajuću arhivu u taj direktorij (stvara se struktura u kojoj se moraju nalaziti datoteke sniff_core.py, sniff_dnscache.py, sniff_filters.py, sniff_sensor.py kao i scapy poddirektorij sa Scapy modulima),
- postaviti izvršne ovlasti nad datotekama sniff_dnscache.py i sniff_sensor.py.

Ovime je instalacija uspješno završena.

9. Dodatak C: Upute za korištenje

Nakon instalacije potrebno je prilagoditi konfiguraciju svojim potrebama:

- ako je riječ o senzoru promjene se unose u datoteku sniff_sensorrc te je moguće prilagoditi sljedeće parametre:
  - **loglevel**: određuje količinu informacija koje se bilježe, npr. "INFO" za bilježenje svih dostupnih informacija, odnosno "CRITICAL" za bilježenje samo kritičnih sistemskih informacija,
  - **logfile**: određuje datoteku u koju se bilježe sistemski događaji, obrađeni DNS paketi i slično, a to je tipično "sniff_sensor.log",
  - **pcapexpr**: obično nije potrebno mijenjati, a standardno je definiran kao "port 53" odnosno sav DNS promet koji dolazi i odlazi s TCP i UDP portova 53; koristeći ovaj PCAP-kompatibilni izraz moguće je preciznije odrediti promatrani promet,
  - **srvaddr**: IPv4 adresa udaljene centralne jezgre,
  - **srvport**: port na kojem udaljena centralna jezgra osluškuje pakete, a to je obično port "5000",
  - **cryptokey**: dijeljena lozinka za enkripciju paketa,
  - **standalone**: zastavica koja definira da senzor radi samostojeće (dakle i analizu DNS prometa) ili to obavlja udaljena jezgra; tipično je "False",
- ako je riječ o jezgri promjene se unose u datoteku sniff_corerc, te je moguće prilagoditi sljedeće parametre:
  - **loglevel**: određuje količinu informacija koje se bilježe, npr. "INFO" za bilježenje svih dostupnih informacija, odnosno "CRITICAL" za bilježenje samo kritičnih sistemskih informacija,
  - **logfile**: određuje datoteku u koju se bilježe sistemski događaji, obrađeni DNS paketi i slično, a to je tipično "sniff_core.log",
  - **addr**: IPv4 adresa na kojoj jezgra osluškuje udaljene pakete,
  - **port**: port na centralna jezgra osluškuje pakete, a to je obično port "5000",
  - **cryptokey**: dijeljena lozinka za enkripciju paketa,

Za uspješno korištenje je nužno i dovoljno sljedeće:

- ako je riječ o senzoru:
  - prilagoditi datoteku sniff_sensorrc na već opisani način,
  - pokrenuti sniff_sensor.py te opcionalno ga poslati u pozadinski način rada,

    - zapisnici o radu se standardno mogu pregledavati u sniff_sensor.log (osim ako nije konfigurirano drukčije),
- ako je riječ o jezgri:
    - prilagoditi datoteku sniff_corerc na već opisani način,
    - pokrenuti sniff_core.py te opcionalno ga poslati u pozadinski način rada,
    - zapisnici o radu se standardno mogu pregledavati u sniff_core.log (osim ako nije konfigurirano drukčije).

Program radi dok se eksplicitno ne ugasi: korištenjem Ctrl-C u interaktivnom (nepozadinskom) načinu rada ili slanjem odgovarajućeg signala za prekid.