Zbornik radova Fakulteta tehničkih nauka, Novi Sad

**UDK: 004.42:004.738.12**
**DOI: https://doi.org/10.24867/18BE08Mimica**

# KONFIGURACIJA REVERZNOG PROKSIJA

# CONFIGURATION OF REVERSE PROXY

Lara Mimica Kostović, *Fakultet tehničkih nauka, Novi Sad*

**Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

**Kratak sadržaj –** *U ovom radu predstavljena je tehnologija reverznih proksi servera kao i konfiguracija istih. Naglasak je na njihovim osnovnim principima, mogućnostima primene, prednostima i nedostacima. U okviru rada posebno je istaknuta njihova arhitektura, kao i primena u preusmeravanju mrežnog prometa prema opterećenju. Kao primer konfiguracije i testiranja opisana je implementacija sistema za praćenje vrednosti ulaganja u kriptovalute. Prilikom implementacije korišćene su sledeće tehnologije: Node.js, Nest.js i TypeScript. Za implementaciju reverznog proksi servera korišćen je NGINX.*

**Ključne reči:** *Reverzni proksi, Server, Konfiguracija*

**Abstract** – *This paper presents the technology of reverse proxy servers as well as their configuration. The emphasis is on their basic principles, possibilities of application, advantages and disadvantages. The paper also highlights their architecture, as well as their application in Load Balancing. The implementation of a system for monitoring the value of investments in cryptocurrencies is described as an example of configuration and testing. The following technologies were used during the implementation: Node.js, Nest.js and TypeScript. NGINX was used to implement the reverse proxy server.*

**Keywords:** *Reverse proxy, Server, Configuration*

## 1. UVOD

U prvom delu rada opisan je koncept rezervnih proksi servera. Kako bi se dodatno razjasnilo funkcionisanje rezernog proksi sistema, sledeća poglavlja detaljno opisuju tok komunikacije između pojedinih komponenti sistema i način testiranja njegove funkcionalnosti. U drugom poglavlju opisana su osnovna načela, kao i prednosti i nedostaci reverznih proksi servera. U trećem poglavlju istaknut je detaljan opis njihovog načina rada. Nešto više o njihovoj primeni kada je u pitanju preusmeravanje mrežnog prometa prema opterećenju moći će da se pročita u četvrtom poglavlju, dok će u petom biti opisana konfiguracija, implementacija kao i mogućnosti testiranja sistema. Poslednje poglavlje obuhvata zaključak rada.

**NAPOMENA:**
**Ovaj rad proistekao je iz master rada čiji je mentor bio dr Željko Vuković, docent.**

## 2. OSNOVNA NAČELA

Reverzni proksi je termin koji opisuje način korišćenja klasičnih proksi servera u drugačijem kontekstu. Za razliku od klasičnih forward proxy servera, koji se ponašaju kao posrednici za konekcije od strane klijenta prema serveru, reverse proxy tehnologija koristi obrnuti pristup, odnosno ovi serveri se ponašaju kao posrednici za konekcije od strane servera ka klijentu, pa tako i dolazi sam naziv reverzni proksi server. Tehnologija reverznih proksi servera u današnje vreme postaje sve popularnija. Dodatni nivo sigurnosti za interne resurse, mogućnost preusmeravanja prometa prema opterećenju (engl. Load Balancing) i prikrivanje informacija o internoj organizaciji računarske infrastrukture samo su neke od njenih prednosti [1].

Budući da se reverse proxy tehnologija u svojoj osnovi velikim delom bazira na načinu rada klasičnih forward proxy servera, uvodni deo rada posvećen je upravo njima. Ukratko su opisana osnovna načela rada i razlozi njihovog korišćenja.

Klijentovo iniciranje konekcije prema serveru na Internetu predstavlja prvi korak i početak komunikacije. Upit klijenta se zatim prosleđuje odgovarajućem proksi serveru, a u zavisnosti od toga da li se radi o transparentnom proksiju ili ne, klijent može, a ne mora biti svestan njegovog postojanja.

Transparentni proksi sistemi danas su puno praktičniji i bolje prihvaćeni, budući da, osim što olakšavaju administraciju sistema, korisnicima omogućavaju ugodniji rad [1].

Proksi server zatim analizira zahtev klijenta, pa u svojoj cache bazi proverava da li postoje sadržaji koji će zadovoljiti upit. Ukoliko takvi sadržaji postoje, klijentu se vraća zatraženi sadržaj i komunikacija se ovde završava.

Ukoliko u cache bazi ne postoje zatraženi sadržaji, upit se dalje prosleđuje Internet serverima kojima je upit bio i izvorno upućen (korak 2).

Server obrađuje primljeni zahtev i nakon obrade ga vraća proksi serverima (korak 3).

Primljeni odgovor proksi zatim prosleđuje klijentu koji je inicirao upit (korak 4), pri čemu u svojoj bazi na određeno vreme kešira procesirane sadržaje (engl. Caching).

Privremeno keširanje sadržaja je jedna od najvećih prednosti korišćenja HTTP proksi servera, budući da se na taj način korisnicima može u velikoj meri povećati kvalitet usluge. Osim mogućnosti privremenog keširanja sadržaja, proksi tehnologija nudi i brojne druge prednosti kao što su proveravanje i filtriranje saobraćaja na temelju

sadržaja paketa (engl. Content Filtering), autentifikacija korisnika i sl [1].

### 2.1 Prednosti i nedostaci reverznih proksi servera

Jedna od najvećih prednosti korišćenja reverznih proksija je mogućnost uspostave centralne tačke pristupa svim internim serverima.

Udaljeni korisnici pristupaju proksi serveru, koji zatim njihove upite preusmerava na interne resurse. Proksi u ovom slučaju funkcioniše poput gateway-a koji pored usmeravanja saobraćaja ima bitan zadatak: unosi dodatan nivo zaštite. On kontroliše saobraćaj koji dolazi sa javne mreže, tj. Interneta.

Ukoliko se na nivou proksi servera koristi detekcija neovlašćenih aktivnosti (engl. Intrusion Detection System), omogućuje se detekcija neovlašćenih aktivnosti usmerenih prema internim sistemima, što samim tim omogućava i njihovo blokiranje.

Još jedna od prednosti korišćenja reverznih proksi servera, koju treba napomenuti, vezana je za mogućnost preusmeravanja prometa na interne sisteme prema njihovoj opterećenosti (engl. Load Balancing). Ukoliko se radi o opterećenim serverima koji svakodnevno primaju velik broj upita, ova mogućnost znatno poboljšava performanse i vreme održivosti sistema. Proksi server zavisno od opterećenja internih servera preusmerava promet tako da se ostvari što bolje vreme odziva za korisnika (engl. Response time) [1].

Budući da je proksi server zadužen za prosleđivanje svih upita prema internim serverima, ovakva konfiguracija sistema omogućava i njihovu jednostavniju zamenu pa i bezbolnije promene u Domain name system (DNS).

U slučaju kvara na nekom od internet sistema, odnosno u slučaju potrebe za promenom DNS-a, jednostavnim modifikacijama na samom proksi serveru moguće je u kratkom roku definisati novu konfiguraciju koja će odgovarati privremenom stanju, dok se ne uklone problemi.

Kao i svaka druga tehnologija, uz svoje prednosti, reverzni proksiji imaju i nedostatke.

Jedan od osnovnih nedostataka je da ukoliko dođe do kvara samog proksija, svi ostali servisi sa kojima je povezan postaju nedostupni (ovo se može rešiti uvođenjem redundantnih proksi servera).

Drugi nedostatak vezan je za sigurnosni rizik koji se javlja ukoliko neovlašćeni korisnik preuzme kontrolu nad proksi serverom. Ovaj problem dolazi još više do izražaja ukoliko je sigurnosna politika firewall-a površno implementirana, ili ukoliko interni serveri nisu adekvatno zaštićeni [1].

Upravo je iz tog razloga od važnosti voditi računa o redovnoj administraciji i instalaciji sigurnosnih delova na svim komponentama koje čine reverzni proksi sisitem, kako bi se na taj način maksimalno sprečile moguće nelagodnosti.

Ukoliko se radi o Web servisu, interni server može biti otvoren za napade koji se šire putem HTTP protokola, bez obzira što se istima pristupa putem proksi servera. Kako bi se otežala mogućnost kompromitovanja internih resursa, s obzirom na jednostavno prosleđivanje konekcija s javnog interneta prema internim serverima, bilo bi poželjno na sam proksi server ugraditi podršku za analizu sadržaja mrežnih paketa (engl. Content Filtering).

Na taj način bi bilo moguće u određenoj meri razlikovati legitimni od nelegitimnog prometa i samim tim donositi odluke o tome da li će se promet proslediti prema internim serverima ili ne.

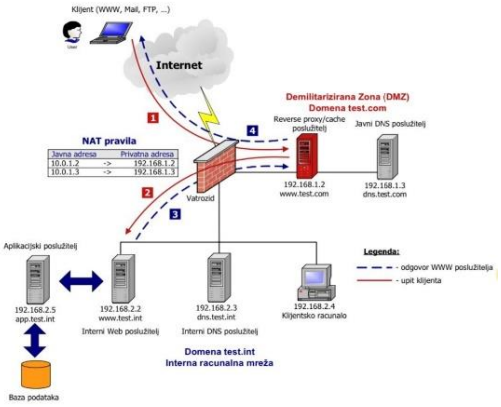
## 3. NAČIN RADA

U ovom poglavlju će biti opisan način rada reverznih proksi servera, osnovna načela reverzne proksi tehnologije s pripadajućim grafičkim prikazima, kao i mogućnosti njene primene u praksi.

Postoje dva osnovna modela za korišćenje servera. Jedan je vezan za osiguravanje internih servera putem proksi servisa, dok je drugi vezan za mogućnosti balansiranja mrežnog prometa.

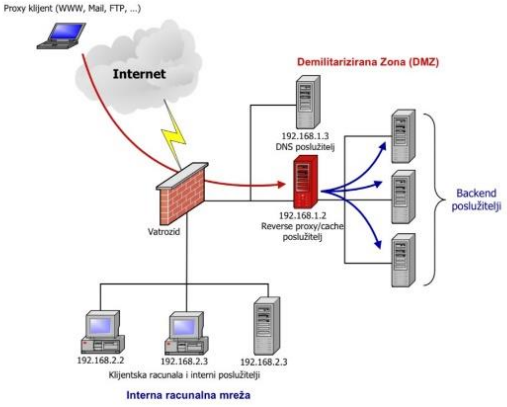
### 3.1 Arhitektura sistema

Na slici 1 prikazan je jedan od mogućih scenarija upotrebe RP tehnologije u svrhu zaštite internih servera, a u radu je detaljno i opisan.




Slika 1. *Primer korišćenja* [1]

Na slici 2 je prikazana arhitektura sa proksi i backend faktorima u DMZ zoni.




Slika 2. *Primer korišćenja sa proksijem i backend faktorima u DMZ zoni* [1]

### 3.2 Komunikacioni tok

Kako bi se omogućio ispravan rad reverznog proksi servera potrebno je definisati sledeća pravila:

- Regularno prepisivanje (engl. Regular mapping) - pravilima regularnog prepisivanja definiše se na koje će se interne servere upiti klijenata prosleđivati.
- Reverzno prepisivanje (engl. Reverse Mapping) - ovim pravilima omogućuje se prikrivanje stvarnih adresa internih backend sistema.

Ovo je vrlo važna karakteristika reverznog proksi servera, budući da upravo ona omogućuje prikrivanje interne strukture računarskog sistema i servisa. Uz definisanja pravila reverznog prepisivanja, proksi server će presretati sve odgovore internih servera pa ih modifikovati na način takav da klijentima izgleda kao da odgovor dolazi s proksija, a ne s internih servera.

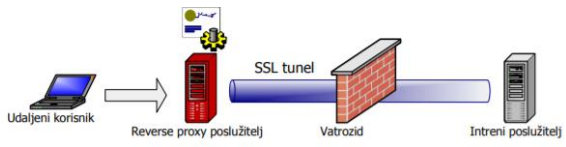
Tok komunikacije je sledeći:

1. Korisnik inicira konekciju prema serveru www.test.com
2. Javni DNS server u DMZ zoni klijentu vraća javnu IP adresu putem koje je dostupan server pod imenom www.test.com (10.0.1.2)
3. Klijent inicira konekciju prema serveru s IP adresom 10.0.1.2, koja preko statičkog prepisivanja adresa na firewall-u prosleđuje sve proksi serveru u DMZ zoni (korak 1)
4. Reverzni proksi server na temelju definisanih regularnih pravila prepisivanja otvara konekciju prema internom serveru (korak 2)
5. Interni server, nakon procesiranja upita, odgovor vraća reverznom proksi serveru u DMZ zoni (korak 3)
6. Proksi server analizira primljeni odgovor, pa ga na osnovu definisanih reverznih pravila modifikuje (ukoliko je to potrebno)
7. Proksi server klijentu vraća odgovor, pri čemu svi delovi odgovara ukazuju na poslati upit

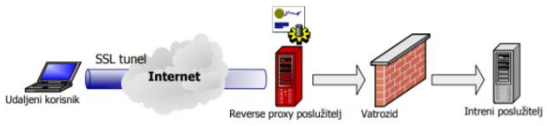
### 3.3 Bezbednost

Ukoliko je potrebno osigurati poverljivost podataka koji se razmenjuju između klijenta i sistema, na reverznom proksi serveru moguće je uključiti i podršku za Secure Socket Layer (SSL) protokol.

U ovakvom scenariju moguća su dva slučaja prikazana na slikama 3 i 4.




Slika 3. *Enkripcija i autentifikacija saobraćaja između RP servera i internih servera* [1]




Slika 4. *Enkripcija i autentifikacija saobraćaja između klijenta i servera* [1]

## 4. PREUSMERAVANJE MREŽNOG PROMETA PREMA OPTEREĆENJU

Još jedna od primena u kojoj se reverzna proksi tehnologija pokazala kao vrlo praktična jeste preusmeravanje mrežnog prometa prema opterećenju (engl. Load Balancing). U ovom slučaju proksi sistem presreće zahteve klijenata te ih u zavisnoti od opterećenja prosleđuje na jedan od internih backend servera (Slika 7) [1].
Tehnika preusmeravanja mrežnog prometa najčešće se primenjuje kod servisa koji konstantno primaju velik broj upita i gde se posebno mora voditi računa o performansama, odnosno vremenu odziva (engl. Response time).

## 5. KONFIGURACIJA I TESTIRANJE

U ovom delu rada opisana je implementacija reverznog proksi servera pomoću NGINX-a. Implementacija reverznog proksija je izvršena na aplikaciji za praćenje vrednosti ulaganja u kriptovalute čija je specifikacija navedena u nastavku:

Prvi deo

1. Implementirati reverzni proksi server pomoću NGINX-a

Drugi deo

1. Neophodno je kreirati entitet Investicija sa poljima Naziv (npr. Bitcoin), Skraćenica (BTC), datum, cena po jedinici (npr. 30.171,43 €), iznos (npr. 0,001), vrednost (= cena po jedinici * iznos). Takođe je potrebno implementirati CRUD (engl. Create Read Update Delete) operacije za svaku investiciju tako da korisnik može dodati investicije u svoj portfolio. Brisanje treba da bude logičko i svi podaci treba da ostanu u bazi podataka.
2. Dodati entitet Portfolio koji prikuplja sve investicije za korisnika.
3. Svakog sata potrebno je čitati vrednosti kriptovaluta iz CointMarketCap API-ja i izračunavati ukupnu vrednost investicionog portfolia (zbir svih iznosa * njihova trenutna vrednost).
4. Sačuvati izračunatu vrednost u entitetu Vrednost portfolia tako da korisnik može da prati vrednosti svog portfolia istorijski
5. Koristiti TypeScript

Treći deo

1. Koristiti HttpModule umesto fetch/axios-a, da bi se vrednosti kriptovaluta mogle sačuvati kao promenljiva koja se može posmatrati (observable)
2. Kreirati servis koji će biti subscribe-ovan na promenljivu koja će se posmatrati. Prilikom svake promene te promenljive, ovaj servis treba da uzme podatke i da ih prosledi klijentu pomoću websocket-a
3. Napraviti jednostavnu klijentsku aplikaciju koja će biti povezana sa serverom pomoću websocket-a. Ova aplikacija će prikazati najnovije vrednosti kriptovaluta koje dolaze sa servera

Prilikom implementacije korišćene su sledeće tehnologije: Node.js, Nest.js i TypeScript.

Uslovi neophodni za realizaciju implementacije su sledeći:

- Linux server
- Korisnik sa sudo privilegijama
- Linux Command-Line (CLI) ili terminal

Konfiguracija i implementacija sistema detaljno je opisana u radu u poglavlju **Proces konfiguracije i implementacije**. Nakon pomenutog poglavlja, u poglavlju **Testiranje sistema** opisana su dva načina za testiranje. Jedan od njih je svakako pristupanje log datoteci sa reverse proxy servera, gde je moguće pronaći podatke o prosleđivanju zahteva sa klijenta na interni server. Takođe, u log datotekama servera moguće je proveriti adresu s koje pristižu HTTP upiti.

## 6. ZAKLJUČAK

U radu je opisana tehnologija reverznog proksi servera zajedno s njenim osnovnim karakteristikama, prednostima, nedostacima, kao i mogućnostima upotrebe. Na kraju je opisan primer implementacije reverznog proksi sistema pomoću NGINX-a sa osnovnim smernicama kojih se treba pridržavati prilikom implementacije sistema ovog tipa. Takođe je analiziran i tok komunikacija između pojedinih komponenti sistema, kao i primer praćenja istog, čime se dodatno demonstrirao način rada sistema.

## 7. LITERATURA

[1] Hrvatska akademska i istraživačka mreža CARNet, 2003. Reverzni proksi poslužitelji
[2] Enrique Ortiz, C. 2002. Introduction to OTA Application Provisioning http://developers.sun.com/techtopics/mobility/midp/articles/ota
[3] Sharad Chandra Agrawal 2003. Location Based Service http://www.tcs.com/0_whitepapers/htdocs/atc/location_based_services_sep03.pdf
[4] Marin Vuković 2006. Isporuka lokacijski specifičnog sadržaja pokretnim korisnicima, diplomski rad, Fakultet elektrotehnike i računarstva, Zagreb
[5] Bažant, Lovrek, Mikac i drugi 2003. Osnovne arhitekture mreža Element, Zagreb
[6] (2012) Technet - Echange Server 2010. Dostopna na: http://technet.microsoft.com/en-US/exchange/dd203064
[7] T. Redmond, Microsoft Exchange Server 2010 Inside Out, Microsoft Press, 2010
[8] J. McBee, D. Elffasy, Mastering Microsoft Exchange Server 2010, Sybes, 2010

### Kratka biografija

**Lara Mimica Kostović** rođena je 01. februara 1998. godine u Novom Sadu. Godine 2016. upisala je Fakultet tehničkih nauka, odsek Računarstvo i automatika. Oktobra 2020. godine je diplomirala. Iste godine upisala je master studije na Fakultetu tehničkih nauka u Novom Sadu, odsek Računarstvo i automatika, studijski program Elektronsko poslovanje. Master rad odbranila je 2021. godine.