HRVATSKA AKADEMSKA I ISTRAŽIVAČKA MREŽA
CROATIAN ACADEMIC AND RESEARCH NETWORK

# Reverzni *proxy* poslužitelji

**CCERT-PUBDOC-2003-05-21**

CARNet CERT u suradnji s LS&S

Sigurnosni problemi u računalnim programima i operativnim sustavima područje je na kojem CARNet CERT kontinuirano radi.

Rezultat toga rada ovaj je dokument koji je nastao suradnjom CARNet CERT-a i LS&S-a, a za koji se nadamo se da će Vam koristiti u poboljšanju sigurnosti Vašeg sustava.

**CARNet CERT**, www.cert.hr - nacionalno središte za **sigurnost računalnih mreža** i sustava.

**LS&S**, www.lss.hr - laboratorij za sustave i signale pri Zavodu za elektroničke sustave i obradbu informacija Fakulteta elektrotehnike i računarstva Sveučilišta u Zagrebu.

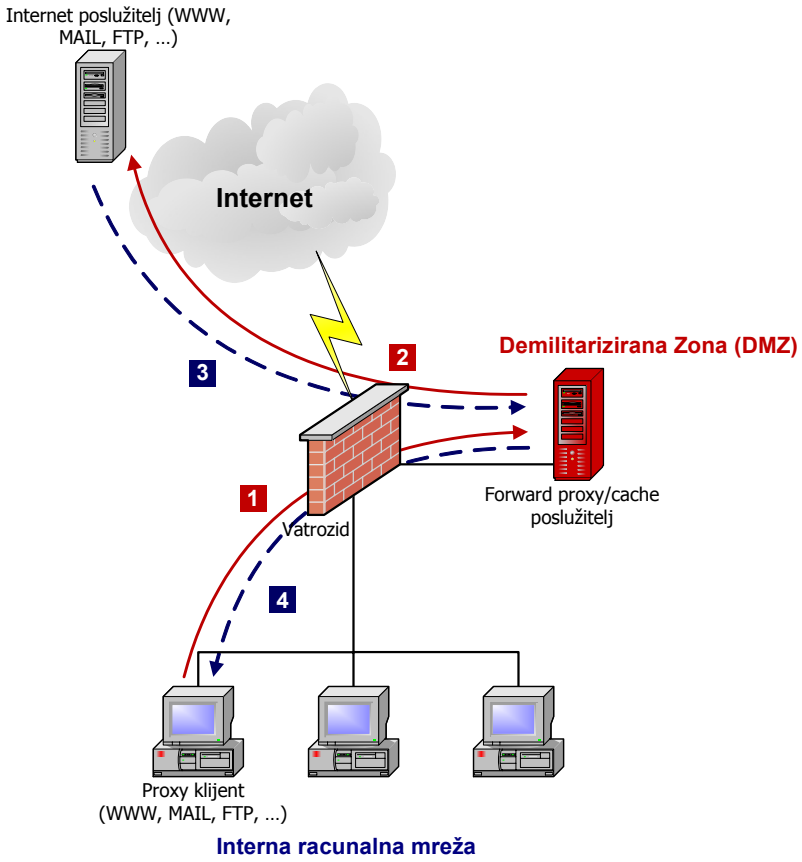
# Sadržaj

# 1. Uvod

U ovom dokumentu biti će opisana tehnologija reverznih *proxy* (engl. *Reverse proxy*) poslužitelja, arhitektura koja osnovni koncept preuzima od klasičnih *proxy* poslužitelja, ali u nešto drukčijem kontekstu. Osnovna načela, mogućnosti primjene te prednosti i nedostaci korištenja reverse *proxy* tehnologije biti će pokriveni većim dijelom dokumenta.
Koncept reverznih *proxy* poslužitelja, opisan u prvom dijelu dokumenta, u Poglavlju 6 je primijenjen na konkretnom primjeru. Korištenjem Apache Web poslužitelja implementiran je *reverse proxy* sustav na kojem su demonstrirani osnovni postupci uspostave i konfiguracije. U svrhu dodatnog pojašnjena načina rada *reverse proxy* sustava, u narednim je poglavljima detaljno opisan tijek komunikacije između pojedinih komponenti sustava te način testiranja njegove funkcionalnosti.

# 2. Osnovna načela

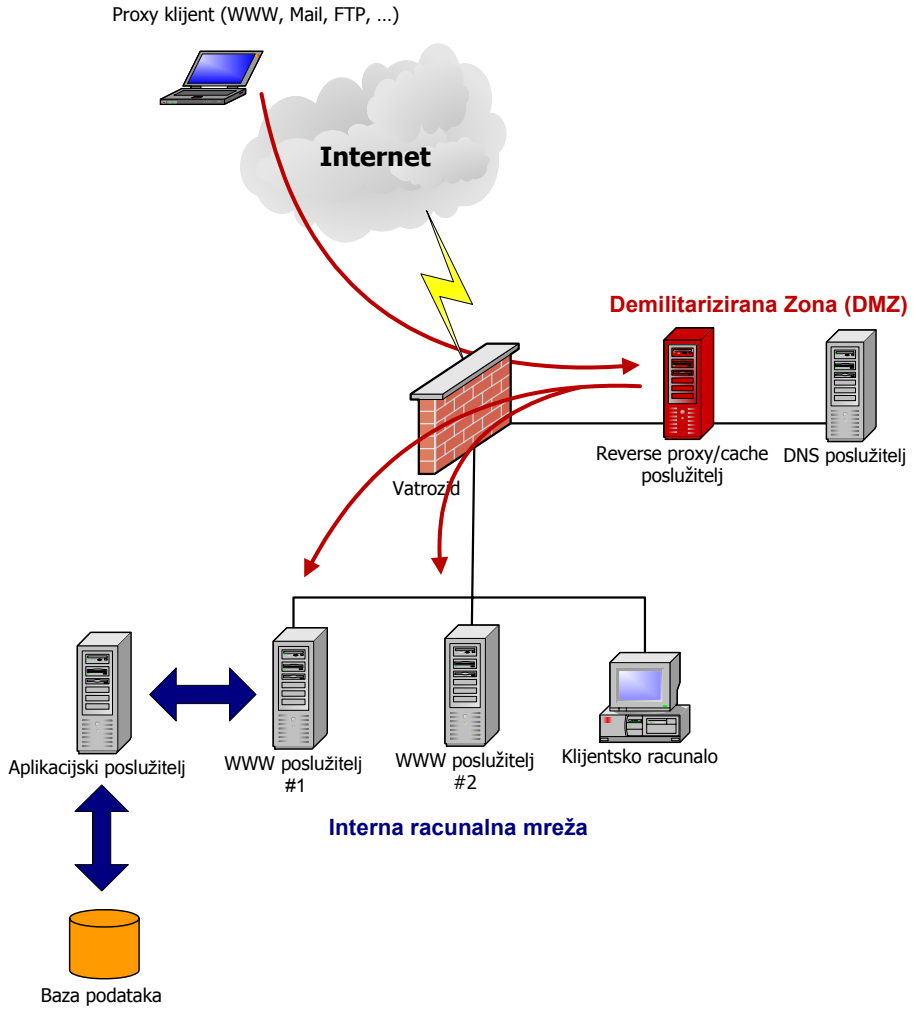
Reverzni *proxy* je termin koji opisuje način korištenja klasičnih *proxy* poslužitelja u nešto drukčijem kontekstu. Za razliku od klasičnih *forward proxy* poslužitelja (***Slika 1***), koji se ponašaju kao posrednici za konekcije inicirane od strane klijenta prema poslužitelju, *reverse proxy* tehnologija koristi obrnuti pristup, otkuda i dolazi sam naziv reverzni *proxy* poslužitelj.
Tehnologija reverznih *proxy* poslužitelja u današnje vrijeme postaje sve popularnija. Dodatni nivo sigurnosti za interne resurse, mogućnost preusmjeravanja prometa prema opterećenju (engl. *Load balancing*) i prikrivanje informacija o internoj organizaciji računalne infrastrukture samo su neke od njenih prednosti.
Čitatelji koji su upoznati sa tehnologijom klasičnih *proxy* poslužitelja dobro znaju da se oni mogu primjenjivati na različite Internet servise (Web (HTTP), FTP, Mail (SMTP) i sl.). Isto pravilo vrijedi i za *reverse proxy* poslužitelje, s time da, bez obzira o kojem se servisu radi, osnovna načela ostaju ista. U nastavku dokumenta naglasak će biti dan isključivo na Web (HTTP) *proxy* poslužiteljima, s obzirom da je to trenutno njihova najraširenija primjena.
Budući da se *reverse proxy* tehnologija u svojoj osnovi velikim dijelom bazira na načinu rada klasičnih *forward proxy* poslužitelja, uvodni dio dokumenta biti će posvećen upravo njima. Biti će ukratko opisana osnovna načela rada te razlozi njihovog korištenja.
U prvom koraku komunikacije klijent inicira konekciju prema poslužitelju na Internetu (korak 1). Ova konekcija automatski se prosljeđuje odgovarajućem *proxy* poslužitelju na obradu, a ovisno o tome da li se radi o transparentnom *proxy* sustavu ili ne, klijent može, a i ne mora biti svjestan njegovog postojanja. Transparentni *proxy* sustavi danas su puno praktičniji i bolje prihvaćeni, budući da, osim što olakšavaju administraciju sustava, korisnicima omogućuju i ugodniji rad.
Pri primanju upita *proxy* poslužitelj analizira zahtjev klijenta, te u svojoj *cache* bazi provjerava da li postoje sadržaji koji će zadovoljiti upit. Ukoliko takvi sadržaji postoje (engl. *Cache hit*), klijentu se iz *cache* baze vraća zatraženi sadržaj i komunikacija ovdje završava. Ukoliko u *cache* bazi ne postoje zatraženi sadržaji (engl. *Cache miss*), upit se dalje prosljeđuje Internet poslužitelju kojemu je bio i izvorno upućen (korak 2).
Poslužitelj obrađuje zahtjev te ga nakon obrade vraća *proxy* poslužitelju (korak 3). Ovaj odgovor *proxy* zatim prosljeđuje klijentu koji je inicirao upit (korak 4), pri čemu u svoj *cache* bazi na određeno vrijeme pohranjuje procesirane sadržaje.
Privremeno pohranjivanje sadržaja (engl. *Caching*), jedna je od najvećih prednosti korištenja HTTP *proxy* poslužitelja, budući da se na taj način može u velikoj mjeri smanjiti količina prometa između interne računalne mreže i Interneta, a ujedno se i korisnicima pruža bolja kvaliteta usluge.
Osim mogućnosti privremenog pohranjivanja sadržaja, *proxy* tehnologija nudi i brojne druge prednosti kao što su provjeravanje i filtriranje prometa na temelju sadržaja paketa (engl. *Content filtering*), autentikacija korisnika i sl.
**Napomena:** U danom primjeru *proxy* poslužitelj nalazi se u Demilitariziranoj zoni (engl. *Demilitarized Zone – DMZ*), implementiranoj putem vatrozida. Treba napomenuti da je to samo jedan od mogućih načina organizacije *proxy* servisa i da će konkretne realizacije uvelike ovisiti o strukturi ostatka računalne infrastrukture i zahtjeva koji se postave pred sustav.




*Slika 1 Forward proxy poslužitelj*

Nakon kratkog opisa klasičnih *forward proxy* poslužitelja u nastavku slijedi slično razmatranje i za reverzne *proxy* sustave (***Slika 2***).

Sa priložene slike može se vidjeti značajna razlika u odnosu na prethodni slučaj. U ovom slučaju *proxy* poslužitelj prihvaća konekcije vanjskih korisnika te ih dalje prosljeđuje internim poslužiteljima (engl. *Backend server*), prema definiranim pravilima. Ovakva arhitektura internim resursima pruža dodatni nivo zaštite od neovlaštenih aktivnosti koje prijete s Interneta, budući da vanjski korisnici internim servisima pristupaju isključivo putem *proxy* servisa.




*Slika 2: Reverse proxy poslužitelj*

Kombiniranjem SSL (engl. ***S**ecure **S**ocket **L**ayer*) protokola s mogućnostima reverznih *proxy* poslužitelja moguće je djelomično ostvariti slične rezultate kao i prilikom korištenja VPN tehnologije. Upotreba SSL protokola omogućuje enkripciju prometa putem RSA algoritma i autentikaciju korisnika putem certifikata, dok upotreba reverznog *proxy* poslužitelja omogućuje implementaciju centralnog pristupnog mjesta putem kojeg je moguće pristupati internim resursima.

Kao i u prethodnom slučaju i ovo je samo jedan od mogućih načina organizacije sustava. Ovisno o potrebama, poslužitelji kojima se želi omogućiti pristup mogu se nalaziti ili u DMZ zoni zajedno s *proxy* poslužiteljem, ili se mogu nalaziti na internoj računalnoj mreži, kao što je to slučaj u ovome primjeru. Jednako tako, moguće je i nešto skuplje rješenje u kojem se umjesto jednog vatrozida koriste dva. U takvom scenariju se DMZ zona s reverznim *proxy* poslužiteljem nalazi između dva vatrozida, čime se dodatno podiže sigurnosni nivo cijelog sustava.

Prednosti i nedostaci reverznih *proxy* poslužitelja, zajedno s njihovim mogućnostima, biti će detaljnije opisani u nastavku dokumenta.

## 3. Prednosti i nedostatci reverznih *proxy* poslužitelja

Jedna od najvećih prednosti korištenja reverznih *proxy* poslužitelja je mogućnost uspostave centralne točke pristupa internim resursima. Udaljeni korisnici na ovaj način svim internim resursima pristupaju putem *proxy* poslužitelja, čime se unosi dodatni nivo zaštite između povjerljivih internih resursa i nepouzdane javne mreže, odnosno Interneta.

Na ovaj način također je moguće na jednom centralnom mjestu provoditi kontrolu pristupa internim resursima, čime se smanjuje mogućnost pogreške s obzirom na propuste u administraciji sustava.

Ukoliko se na nivou *proxy* poslužitelja uključi i podrška za detekciju neovlaštenih aktivnosti (engl. *Intrusion Detection System*), ovakvim pristupom omogućuje se pravovremena detekcija neovlaštenih aktivnosti usmjerenih prema internim Web poslužiteljima te njihovo trenutno blokiranje.
Još jedna od prednosti korištenja reverznih *proxy* poslužitelja, koju treba spomenuti, vezana je uz mogućnost preusmjeravanja prometa na interne poslužitelje prema njihovoj opterećenosti (engl. *Load Balancing*). Ukoliko se radi o jače opterećenim poslužiteljima, koji svakodnevno primaju vrlo velik broj upita, ova mogućnost može znatno poboljšati performanse i vrijeme odziva sustava. P*roxy* poslužitelj, ovisno o opterećenju internih poslužitelja, preusmjerava promet tako da se ostvari što bolje vrijeme odziva za korisnika (engl. *Response time*). Više informacija o upotrebi *reverse proxy* poslužitelja u svrhu balansiranja mrežnog prometa dano je u Poglavlju 5.
Budući da je *proxy* poslužitelj zadužen za prosljeđivanje svih upita prema internim poslužiteljima, ovakva konfiguracija sustava omogućuje i njihovu jednostavniju zamjenu te "bezbolnije" promjene u DNS imenima. U slučaju kvara na nekom od internih poslužitelja ili u slučaju potrebe za promjenom DNS imena, jednostavnim modifikacijama na samom *proxy* poslužitelju moguće je, u kratkom roku, definirati novu konfiguraciju, koja će odgovarati privremenom stanju, dok se ne uklone problemi.
Kao i svaka druga tehnologija, tako i *reverse proxy* poslužitelji, uz svoje prednosti, imaju i neke od nedostataka. Jedan od osnovnih nedostataka je taj da će, ukoliko dođe do kvara samog *proxy* poslužitelja, svi ostali servisi koji se na njemu baziraju biti nedostupni (osim ukoliko ne postoji redundantni *proxy* poslužitelj, koji automatski podiže cijenu cijelog sustava). Drugi nedostatak vezan je uz sigurnosni rizik koji se javlja ukoliko neovlašteni korisnik preuzme kontrolu nad *proxy* poslužiteljem. Ovaj problem dolazi još više do izražaja ukoliko je sigurnosna politika vatrozida površno implementirana, ili ukoliko interni *backend* poslužitelji nisu adekvatno zaštićeni.
Upravo je iz tog razloga posebno važno voditi računa o redovitoj administraciji i instalaciji sigurnosnih zakrpi na svim komponentama koje čine *reverse proxy* sustav, kako bi se na taj način maksimalno spriječile moguće neugodnosti.
Iako korištenje RP tehnologije pruža dodatni nivo zaštite za interne poslužitelje, oni još uvijek ostaju ranjivi na napade koji šire putem servisa za koji se koristi *proxy* poslužitelj. Ukoliko se radi o Web servisu, interni će poslužitelji još uvijek biti ranjivi na napade koji se šire putem HTTP protokola, bez obzira što se istima pristupa putem *proxy* poslužitelja. Kako bi se otežala mogućnost kompromitiranja internih resursa, s obzirom na jednostavno prosljeđivanje konekcija s javnog Interneta prema internim poslužiteljima, bilo bi poželjno na sam *proxy* poslužitelj ugraditi podršku za analizu sadržaja mrežnih paketa (engl. *Content filtering*).
Na taj način bilo bi moguće, u određenoj mjeri, razlikovati legitimni od nelegitimnog prometa te donositi odluke o tome da li će se promet proslijediti prema internim poslužiteljima ili ne. Ukoliko je ugrađena podrška za analizu prometa dovoljno kvalitetna, na ovaj način moguće je s vrlo velikom preciznošću detektirati, a samim time i blokirati, velik broj napada koji dolaze s Interneta.

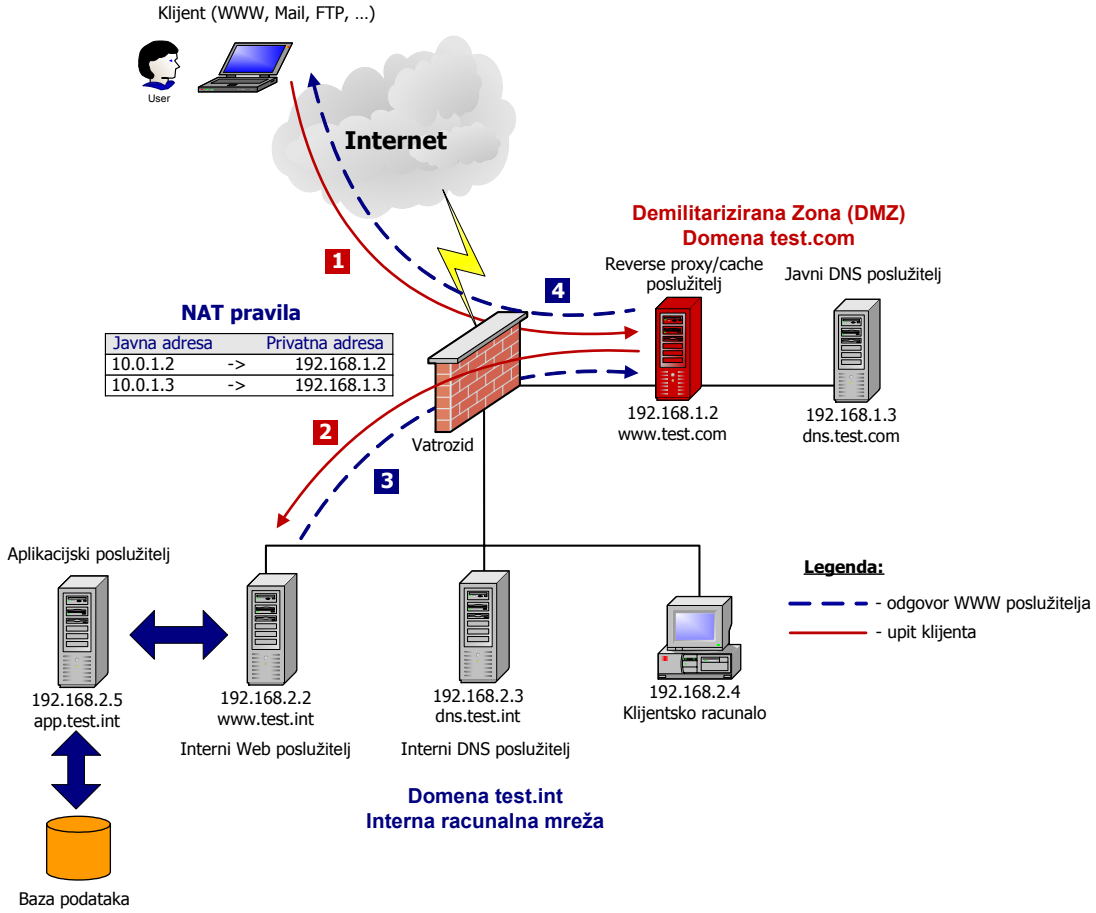
# 4. Način rada

U ovom poglavlju biti će nešto detaljnije opisan način rada reverznih *proxy* poslužitelja. Biti će opisana osnovna načela *reverse proxy* tehnologije, s pripadajućim grafičkim prikazima, te mogućnosti njene primjene u praksi.
Postoje dva osnovna modela za korištenje RP poslužitelja. Jedan je vezan uz osiguravanje internih poslužitelja putem *proxy* servisa, dok je drugi vezan uz mogućnosti balansiranja mrežnog prometa.

## 4.1. Arhitektura sustava

Na sljedećoj slici (
Slika *3*) prikazan je jedan od mogućih scenarija upotrebe RP tehnologije u svrhu zaštite internih poslužitelja, odnosno servisa. U primjeru koji slijedi, RP poslužitelj nalazi se u DMZ zajedno s ostalim javnim poslužiteljima koji su javno dostupni putem statičkog prepisivanja adresa (engl. *Network Address Translation - NAT*). Prepisivanje adresa provodi se na vatrozidu prema sljedećim pravilima:

- 10.0.1.2 -> 192.168.1.2 (www.test.com);
- 10.0.1.3 -> 192.168.1.3 (dns.test.com).



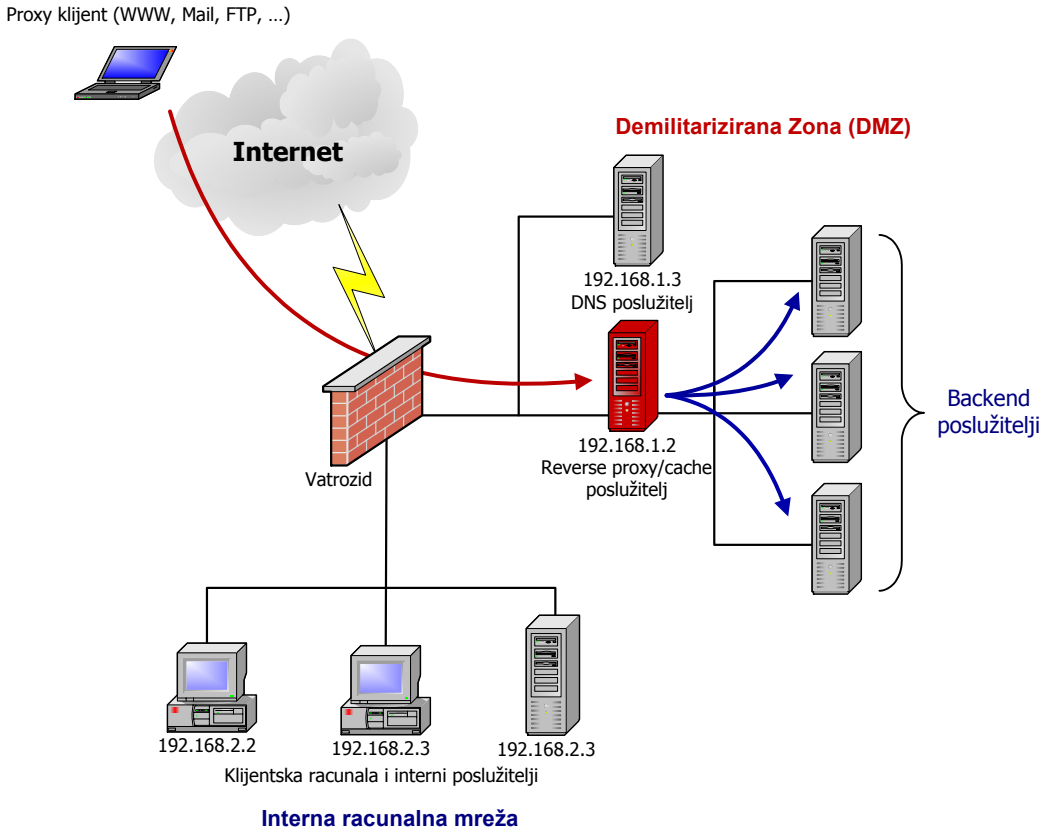

*Slika 3: Primjer korištenja reverse proxy tehnologije*

Reverznom *proxy* poslužitelju (192.168.1.2) je putem javnog DNS (192.168.1.3) poslužitelja pridjeljeno ime www.test.com, koje pokazuje na javnu IP adresu, preko koje je *proxy* poslužitelj dostupan s Interneta. Javni DNS poslužitelj sadrži podatke samo o javnim poslužiteljima, kojima se želi dozvoliti pristup s Interneta. Za dolazak do IP adresa internih backend poslužitelja javni DNS poslužitelj u ovom primjeru koristi statičke zapise u `/etc/hosts` datoteci.

Za razrješavanje imena na internoj računalnoj mreži koristi se interni DNS poslužitelj (192.168.2.3), koji sadrži podatke o računalima u internoj domeni test.int. Internom Web poslužitelju pridjeljeno je ime www.test.int, s pripadajućom IP adresom 192.168.2.2. Web poslužitelj preko aplikacijskog poslužitelja (192.168.2.5) pristupa bazi podataka u kojoj se nalaze podaci bitni za rad Web aplikacije.

Sigurnosnu politiku vatrozida potrebno je definirati tako da se osigura što veći nivo sigurnosti javnih i internih poslužitelja, o čemu će biti više riječi u jednom od narednih poglavlja (Poglavlje 4.3).

Osim upravo opisanog primjera, mogući su i drukčiji primjeri organizacije *reverse proxy* sustava. Jedan od njih je i onaj u kojemu se i *reverse proxy* i *backend* poslužitelji nalaze u DMZ zoni (***Slika 5***). Premještanjem *backend* poslužitelja u DMZ zonu smanjuje se sigurnosni rizik s obzirom na mogućnosti kompromitiranja *proxy* poslužitelja, budući da se u ovom slučaju promet ne prosljeđuje na privatnu računalnu mrežu.




*Slika 4: Arhitektura sa proxy i backend poslužiteljima u DMZ zoni*

## 4.2. Tijek komunikacije

Kako bi se omogućio ispravan rad *reverse proxy* poslužitelja potrebno je definirati dva tipa pravila:

- Regularno prepisivanje (engl. *Regular mapping*) – pravilima regularnog prepisivanja definira se na koje će se interne poslužitelje upiti klijenata prosljeđivati. U primjeru sa slike bilo bi potrebno definirati regularno pravilo kojim će se sav Web promet prema poslužitelju www.test.com preusmjerivati na interni poslužitelj www.test.int. Bez odgovarajućeg regularnog pravila *proxy* poslužitelj neće znati kamo prosljeđivati HTTP/HTTPS upite klijenata.
- Reverzno prepisivanje (engl. *Reverse mapping*) – ovim pravilima omogućuje se prikrivanje stvarnih adresa internih *backend* poslužitelja. Ovo je vrlo važna karakteristika reverse *proxy* poslužitelja, budući da upravo ona omogućuje prikrivanje interne strukture računalnog sustava i servisa. Uz definirana pravila reverznog prepisivanja, *proxy* poslužitelj će presretati sve odgovore internih poslužitelja te ih modificirati na način da klijentima izgleda kao da odgovor dolazi s *proxy*, a ne s internih poslužitelja. Sve elemente HTTP/HTTPS odgovora, koji odaju stvarni izvor paketa, *proxy* će poslužitelj zamijeniti sa svojom IP adresom, odnosno imenom. Kao primjer može se navesti slučaj u kojem interni poslužitelj, u slučaju pogreške, klijenta preusmjerava na neku drugu internu URL adresu. Kako bi klijent i toj novoj adresi pristupio preko *proxy* poslužitelja, a ne izravno, *proxy* poslužitelj, na temelju reverznih pravila, modificira odgovor na način da klijenta usmjeri na svoju IP adresu.

Nakon što su na RP poslužitelju definirana odgovarajuća pravila prepisivanja, moguće je pristupiti internim *backend* poslužiteljima. Tijek komunikacije je sljedeći (
Slika *3*):

- Korisnik inicira konekciju prema poslužitelju www.test.com;
- Javni DNS poslužitelj u DMZ zoni klijentu vraća javnu IP adresu putem koje je dostupan poslužitelj pod imenom www.test.com (10.0.1.2);

- Klijent inicira konekciju prema poslužitelju s IP adresom 10.0.1.2, koja se preko statičkog prepisivanja adresa na vatrozidu prosljeđuje *reverse proxy* poslužitelju u DMZ zoni (korak 1);
- *Reverse proxy* poslužitelj na temelju definiranih regularnih pravila prepisivanja otvara konekciju prema internom WWW poslužitelju (korak 2);
- Interni WWW poslužitelj, nakon procesiranja upita, odgovor vraća *reverse proxy* poslužitelju u DMZ zoni (korak 3);
- *Proxy* poslužitelj analizira primljeni odgovor te ga na temelju definiranih reverznih pravila modificira, ukoliko je to potrebno;
- *Proxy* poslužitelj klijentu vraća odgovor, pri čemu svi dijelovi odgovora ukazuju da je upit procesiran od strane *proxy* poslužitelja.

U sljedećem poglavlju biti će opisani osnovni elementi sigurnosti o kojima je potrebno voditi računa prilikom implementacije *reverse proxy* arhitekture.

## 4.3. Sigurnost

Postoji nekoliko elemenata o kojima posebno treba voditi računa prilikom implementacije *reverse proxy* sustava, a jedna od najvažnijih je sigurnost. Ukoliko je implementacija sustava površna i nedovoljno zaštićena, sve prednosti *reverse proxy* tehnologije (Poglavlje 3) brzo se gube.

Najveću pažnju potrebno je posvetiti definiranju sigurnosne politike vatrozida te odgovarajućim NAT pravilima. Osim stroge sigurnosne politike, korištenje statičkog prepisivanja adresa za javne poslužitelje svakako je preporučljivo, budući da se na taj način smanjuje količina raspoloživih informacija o internoj organizaciji računalne mreže. Budući da je faza prikupljana informacija o ciljnom sustavu (engl. *Information gathering*) jedan od prvih koraka neovlaštenih korisnika prilikom provođenja napada, ovo je samo još jedan od načina da se sustav dodatno zaštiti od neautoriziranog pristupa.

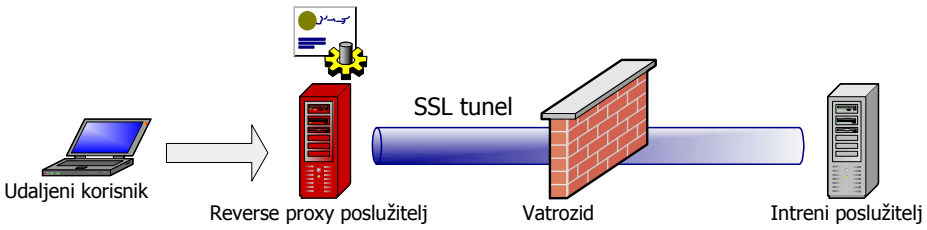
Sigurnosnu politiku vatrozida trebalo bi tako podesiti da su HTTP (TCP/80) i HTTPS (TCP/443) konekcije s javnog Interneta dozvoljene samo na *proxy* poslužitelj u DMZ zoni. Na sličan način potrebno je HTTP i HTTPS konekcije prema internim *backend* poslužiteljima dozvoliti isključivo s *proxy* poslužitelja.

Putem javnog DNS servisa potrebno je osigurati da je samo ime javnog Web poslužitelja dostupno na Internetu (www.test.com). Sva imena internih računala trebaju imati isključivo interni karakter i ne smiju biti javno objavljena na Internetu. Ovakvom konfiguracijom osigurava se da će sav Web promet usmjeren na poslužitelj www.test.com biti procesiran isključivo od strane *proxy* poslužitelja. Ukoliko se na internoj računalnoj mreži koriste javne IP adrese, sve izravne konekcije s javnog Interneta prema njima trebale bi biti blokirane.

Na *proxy* poslužitelju ili vatrozidu moguće je dodatno definirati različite pristupne liste (engl. *access control list*) kojima će se preciznije definirati ovlasti pristupa internim resursima. Ovisno o prioritetu i povjerljivosti podataka moguće je pristup ograničiti na samo neke IP adrese ili HTTP metode, a prema potrebi moguće je provoditi i autentikaciju korisnika.

Ukoliko je potrebno osigurati povjerljivost podataka koji se razmjenjuju između klijenta i poslužitelja, na *reverse proxy* poslužitelju moguće je uključiti i podršku za SSL protokol. U ovakvom scenariju moguća su dva slučaja:

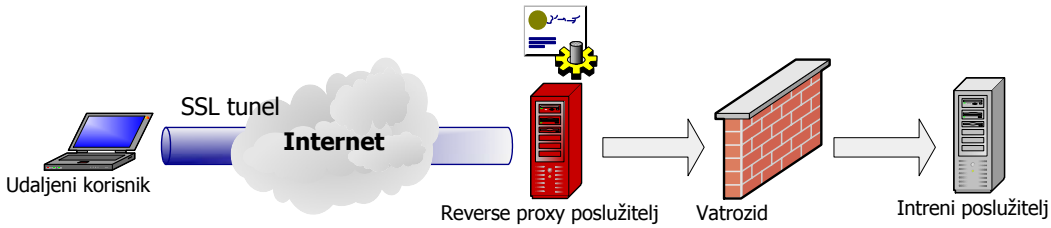
- Enkripcija podataka između RP poslužitelja i internih *backend* poslužitelja;




*Slika 5: Enkripcija i autentikacija prometa između proxy poslužitelja i internih poslužitelja*

Ovakvom konfiguracijom kompletna komunikacija između *proxy* poslužitelja biti će kriptirana i autenticirana putem certifikata.

- Enkripcija podataka između udaljenog klijenta i RP poslužitelja.

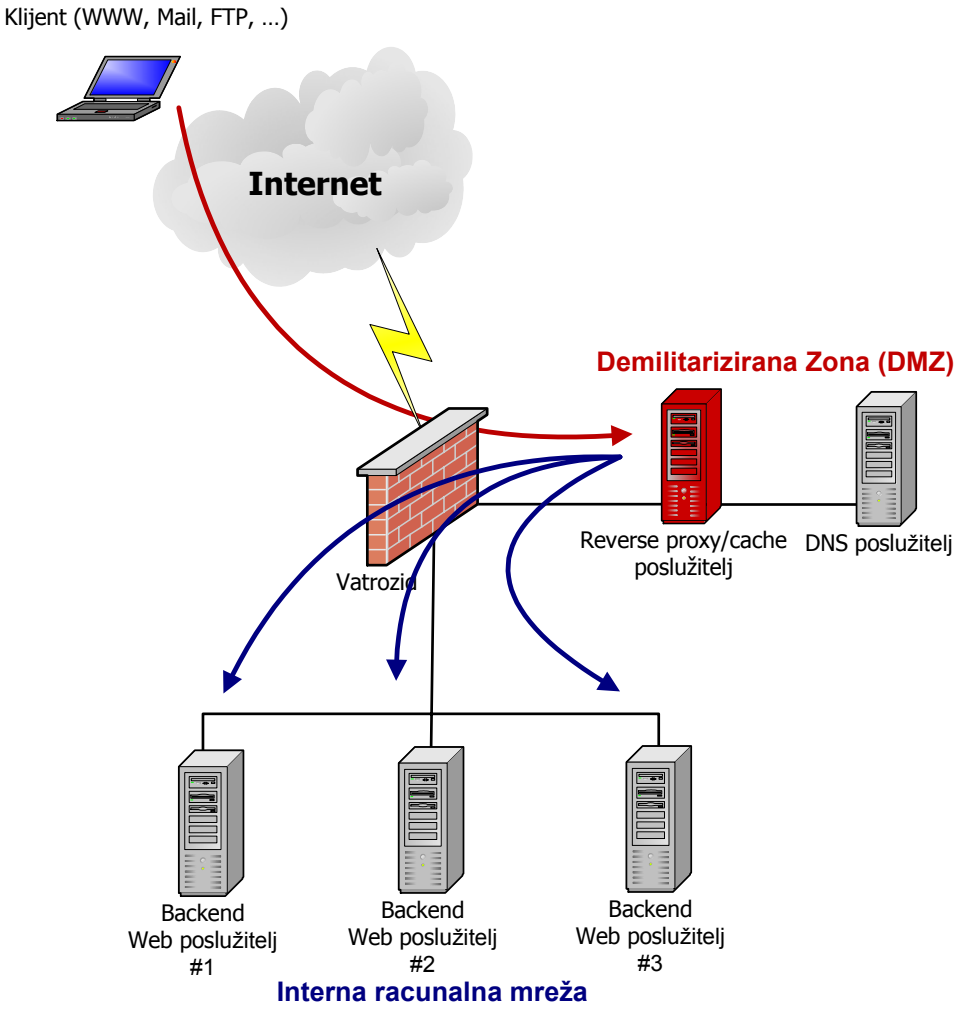



*Slika 6: Enkripcija i autentikacija prometa između klijenta i proxy poslužitelja*

Za razliku od prethodnog primjera ovdje se promet kriptira između udaljenog korisnika i *proxy* poslužitelja. Ovakva arhitektura posebno je zanimljiva budući da štiti mrežni promet koji putuje preko javnog Interneta.
Koji će se od opisanih pristupa koristiti ovisiti će, naravno, o zahtjevima koji se postavljaju pred sustav.

# 5. Preusmjeravanje mrežnog prometa prema opterećenju

Još jedna od primjena u kojima se *reverse proxy* tehnologija pokazala kao vrlo praktična je preusmjeravanje mrežnog prometa prema opterećenju (engl. *Load balancing*). U ovom slučaju *proxy* poslužitelj presreće zahtjeve klijenata te ih, ovisno o opterećenju, prosljeđuje na jedan od internih *backend* poslužitelja (***Slika 7***).
Tehnika preusmjeravanja mrežnog prometa najčešće se primjenjuje kod servisa koji konstantno primaju iznimno velik broj upita i gdje se posebno mora voditi računa o performansama, odnosnu vremenu odziva (engl. *Response time*). Postoji nekoliko načina na koje je moguće riješiti ovaj problem, a *reverse proxy* samo je jedan od njih.
Odluke o preusmjeravanju prometa moguće je donositi na temelju različitih karakteristika. Ukoliko se radi o Web servisu, moguće je, npr., procesorski intenzivnije zadatke (CGI skripte i sl.) proslijediti na posebnu, za to namijenjenu, grupu *backend* poslužitelja, a sve ostale zahtjeve za statičkim Web sadržajima preusmjeriti na preostale poslužitelje. Na ovaj način moguće je optimizirati performanse Web servisa te smanjiti vrijeme odziva.




*Slika 7: Primjena reverse proxy poslužitelja za balansiranje mrežnog prometa*

# 6. Primjer implementacije

*Reverse proxy* sustav moguće je implementirati različitim programskim paketima kao što su:

- Apache Web poslužitelj;
- Squid proxy poslužitelj;
- Microsoft ISA Proxy poslužitelj i sl.

U ovom dokumentu biti će opisan primjer implementacije *reverse proxy* poslužitelja pomoću Apache Web poslužitelja. Biti će opisani osnovni postupci uspostave, konfiguracije i testiranja sustava zajedno s tijekom konekcija između pojedinih komponenti sustava i pripadajućim pojašnjenima.

## 6.1. Instalacija Apache Web poslužitelja

Za kvalitetnu i pouzdanu implementaciju *reverse proxy* sustava baziranog na Apache Web poslužitelju, vrlo važan korak predstavlja sama instalacija i inicijalno podešavanje. S obzirom na kompleksnost i mogućnosti, ovog inače vrlo popularnog programa, dodatnu je pažnju potrebno posvetiti i njegovim sigurnosnim karakteristikama.

Uključivanje potrebnih modula, instalacija odgovarajućih sigurnosnih zakrpi te osnovna konfiguracija koja će zadovoljiti sigurnosne zahtjeve predstavljene pred sustav, temeljni su uvjeti koje je potrebno osigurati prije početka uspostave *reverse proxy* funkcionalnosti.

Kako bi Apache Web poslužitelj mogao obavljati ulogu *reverse proxy* poslužitelja, neophodno je prilikom njegovog prevođenja uključiti `mod_rewrite` i `proxy_module` Apache module. Moduli

mogu biti prevedeni ili kao dinamički (engl. *Dynamic Shared Objects - DSO*) ili kao statički, ovisno o potrebama i odlukama administratora sustava. Koji od modula su statički ugrađeni u Apache poslužitelj moguće je vidjeti zadavanjem sljedeće naredbe:

```
# httpd -l
```

Postupci instalacije i konfiguracije Apache Web poslužitelja neće biti opisivani u ovom dokumentu, budući da je to tema koja izlazi van područja razmatranja. Između ostaloga, na Internetu postoji mnoštvo javno dostupnih dokumenta koji pokrivaju ovu tematiku te se korisnike upućuje na njihovo čitanje. Jedan od kvalitetnijih dokumenta iz ovog područja je dokument objavljen na Web stranicama američke organizacije za nacionalnu sigurnost (***N**ational **S**ecurity **A**gency* - NSA) i dostupan je na URL adresi http://nsa1.www.conxion.com/support/download.htm.

## 6.2. Konfiguracija vatrozida

Sigurnosnu politiku vatrozida potrebno je prilagoditi na način koji će omogućiti implementaciju odabranog *reverse proxy* rješenja. Postupci konfiguracije vatrozida razlikovati će se od proizvođača do proizvođača, pri čemu postoji grupa općenitih pravila koja vrijede za bilo koji od njih.
Na samom početku potrebno je definirati NAT pravila koja će omogućiti pristup *reverse proxy* i ostalim javnim poslužiteljima u DMZ zoni s javnog Interneta. Dopuštene servise treba svesti na minimalan broj koji će zadovoljiti potrebe sustava. Ukoliko se radi samo o HTTP *proxy* servisu, na vatrozidu je dovoljno omogućiti DNS (53/UDP) i HTTP(TCP/80), odnosno HTTPS (TCP/443) mrežne servise.
Sigurnosna politika za komunikaciju između interne mreže i DMZ zone ovisit će o potrebama organizacije. Ukoliko se u DMZ zoni, osim DNS i *proxy* poslužitelja, nalaze i drugi javni servisi (SMTP, FTP, WWW), pravila filtriranja vatrozida trebati će prilagoditi da se omogući njihovo nesmetano funkcioniranje. Prilikom konfiguracije vatrozida uvijek treba imati na umu općenito pravilo kojim se treba minimizirati broj pravila vatrozida koja omogućuju iniciranje konekcija prema privatnoj mreži.
Ukoliko na vatrozidu postoji podrška za analizu sadržaja paketa (engl. *content filtering*) ili detekciju neovlaštenih aktivnosti (engl. *Intrusion detection*), svakako se preporučuje njeno uključivanje.
Još jedan od vrlo važnih elementa prilikom podešavanja vatrozida je optimiziranje i uključivanje podrške za bilježenje log zapisa. Prikladno podešen sustav za praćenje i bilježenje aktivnosti na sustavu pomoći će u detekciji nepravilnosti u radu te potencijalnih malicioznih aktivnosti koje mogu ugroziti sigurnost sustava.

## 6.3. Konfiguracija Apache poslužitelja

Način rada Apache poslužitelja moguće je podešavati uređivanjem `httpd.conf` konfiguracijske datoteke (na nekim Linux distribucijama i `commonhttpd.conf`).
Na samom početku konfiguracijske datoteke potrebno je, osim općenitih parametara (ime i tip poslužitelja, korijenski direktorij i sl.), uključiti i module koji omogućuju *reverse proxy* funkcionalnost (`mod_rewrite` i `proxy_module`).
U nastavku je dan primjer konfiguracijske datoteke u kojoj su navedeni neki općeniti parametri Apache Web poslužitelja i dodani odgovarajući moduli `AddModule` i `LoadModule` direktivama.

```
######################################################################
#                     ***   httpd.conf   ***                       #
#                         Datum: 26/05/03                            #
#    Primjer konfiguracijske datoteke u svrhu implementacije         #
#                 reverse proxy poslužitelja.                        #
######################################################################

# Tip poslužitelja, korijenski direktorij i datoteka s PID
# brojem procesa pod kojim ja Apache poslužitelj pokrenut.

ServerType standalone
ServerRoot "/usr/local/apache"
```

```
PidFile /usr/local/apache/logs/httpd.pid

# Direktive kojima se uključuju potrebni moduli
LoadModule env_module         libexec/mod_env.so
LoadModule config_log_module  libexec/mod_log_config.so
LoadModule mime_module        libexec/mod_mime.so
.
.
.
LoadModule proxy_module libexec/libproxy.so
LoadModule rewrite_module libexec/mod_rewrite.so
.
.
.
<IfDefine SSL>
LoadModule ssl_module         libexec/libssl.so
</IfDefine>
AddModule mod_env.c
AddModule mod_log_config.c
AddModule mod_proxy.c
AddModule mod_rewrite.c
.
.
. <IfDefine SSL>
AddModule mod_ssl.c
</IfDefine>
#  Ime  poslužitelja  i  mrežni  portovi  na  kojima  osluškuje  zahtjeve
#  klijenata i korijenski direktorij za postavljenje Web sadržaje
ServerName www.test.com
DocumentRoot "/home/httpd"

Port 80
<IfDefine SSL>
Listen 80
Listen 443
</IfDefine>
```

Sljedećim direktivama definiraju se inicijalne postavke direktorija kojima Apache Web poslužitelj ima pristup. Ovdje se preporučuju što stroža pravila kako bi se izbjegla ranjivost sustava s obzirom na greške u konfiguraciji. Od ovog trenutka nadalje potrebno je eksplicitno dozvoliti sve potrebne servise odnosno prava pristupa.

```
<Directory />
    Options -FollowSymLinks -SymLinksIfOwnerMatch...
    AllowOverride None
</Directory>
# Ograničenja postavljena na korijenski direktorij za Web sadržaje.
<Directory "/home/httpd">
    Options None
    AllowOverride None
    Order allow,deny
    Allow from all
</Directory>
```

Slijedi definicija datoteka za bilježenje log zapisa.

```
ErrorLog /usr/local/apache/logs/error.log
CustomLog /usr/local/apache/logs/accesslog.combined
```

Slijedećim direktivama definiraju se virtualni Web poslužitelji, čije korištenje omogućuje upotrebu iste IP adrese za više Web *siteova*.

```
NameVirtualHost 192.168.1.2:80
NameVirtualHost 192.168.1.2:443
```

Nakon što su navedeni svi virtualni poslužitelji koji se namjeravaju koristiti, potrebno je posebnim direktivama definirati njihovo ponašanje. Slijedi konfiguracija inicijalnog virtualnog poslužitelja na portu 80, koji je ujedno i poslužitelj koji će biti dostupan s javnog Interneta (www.test.com). Unutar ove `VirtualHost` definicije također će biti navedene i direktive koje uključuju *reverse proxy* funkcionalnost.

```
<VirtualHost 192.168.1.2:80>
    ServerName www.test.com
    DocumentRoot /home/httpd/public
    ErrorLog /usr/log/apache/logs/public_error_log
    CustomLog /usr/log/apache/logs/public_access_log combined
    ProxyRequests on
    RewriteEngine on
    RewriteLog /usr/local/apache/logs/public_rewrite_log
    RewriteLogLevel 1
# Onemogućavanje svih funkcionalnosti unutar ovog virtualnog
# poslužitelja, budući da nisu potrebne
<Directory />
Options None
AllowOverride None
</Directory>
```

Sljedećim direktivama biti će implementirana regularna pravila *proxy* poslužitelja (Poglavlje 4.2), kojima će se upiti Web klijenata prosljeđivati odgovarajućim *backend* poslužiteljima. Definiranje regularnih *proxy* pravila se kod Apache poslužitelja postiže pomoću *rewrite* Apache modula, odnosno direktiva koje on nudi (`RewriteRule`). `Rewrite_mod` Apache modul omogućuje prepisivanje zahtjeva klijenta na temelju regularnih izraza (engl. *regular expressions*) te njihovo prosljeđivanje *backend* poslužiteljima.
Sintaksa korištenja `RewriteRule` direktive je sljedeća:

```
RewriteRule znakovni_niz zamjenski_znakovni_niz [opcije]
```

Na temelju ovako definiranog pravila, Web poslužitelj presreće zahtjeve klijenata te unutar URL polja traži vrijednost `znakovni_niz`. Ukoliko takav niz postoji, biti će zamijenjen znakovnim nizom `zamjenski_znakovni_niz`, nakon čega će se nastaviti s procesiranjem upita. Korištenjem dodatnih opcija moguće je preciznije definirati način na koji će se provoditi prepisivanje upita. U sljedećoj tablici (***Tablica 1***) navedene su neke od značajnijih opcija koje je moguće koristiti u kombinaciji s `RewriteRule` direktivom.

| Opcija | Značenje |
|---|---|
| `-R (redirect)` | Ova opcija klijentu vraća poruku o preusmjeravanju, koja će ga zatim uputiti na novu adresu. `-R` opcija kao argument prihvaća kod HTML poruke koji će biti vraćen klijentu, čime je moguće utjecati na način kako će primljeni odgovor biti protumačen. |
| `-F (forbidden)` | Dodavanjem `-F` opcije klijentu će za navedenu URL adresu biti vraćena poruka `Forbidden` (kod poruke 403). |
| `-P (proxy)` | Upit klijenta se, nakon prepisivanja prema zadanim pravilima, automatski prosljeđuje `proxy` modulu koji će dalje nastaviti s procesiranjem. |
| `NC (nocase)` | Ovom opcijom regularni izrazi neće praviti razliku između velikih i malih slova. |
| `L (last)` | Ovom opcijom se definirano pravilo proglašava zadnjim u lancu. Sva ostala pravila iza njega (ukoliko takva postoje) biti će zanemarena. |

| Opcija | Značenje |
|---|---|
| N<br>(next) | Ovom opcijom se prekida se procesiranje definiranih sljedećih pravila i počinje se iz početka. |

*Tablica 1: Opcije RewriteRule direktive*

U nastavku su dana dva jednostavna primjera, koji demonstriraju način korištenja `RewriteRule` direktive pri implementaciji *reverse proxy* sustava. Navedenim pravilima svi zahtjevi upućeni na adrese www.test.com/dokumentacija i www.test.com/arhiva biti će proslijeđeni na interni *backend* poslužitelj www.test.int.

```
RewriteRule ^/(dokumentacija)/(.*)
http://www.test.int/www_dokumentacija/$1 [NC, P]
RewriteRule ^/(arhiva)/(.*) http://www.test.int/arhiva/$1 [NC, P]
```

Ovisno o specifičnostima implementacije različitih mrežnih sustava, moguće je definirati više pravila, kojima će se zadovoljiti postavljeni zahtjevi.
Upravo opisanim postupkom definiranja regularnih *proxy* pravila pomoću `RewriteRule` Apache direktive, obavljena je prva polovica zadatka pri implementaciji *reverse proxy* rješenja. Za potpunu funkcionalnost sustava potrebno je još definirati i reverzna *proxy* pravila, koja će omogućiti modifikaciju HTTP odgovora *backend* poslužitelja te tako prikriti njihovu prisutnost. Iako postojanje reverznih proxy pravila nije neophodno za osnovno funkcioniranje *reverse proxy* sustava, njihova definicija jednako je važna, budući da je prikrivanje *backend* poslužitelja jedna od temeljnih ideja *reverse proxy* arhitekture.
Reverzna *proxy* pravila moguće je, na nivou Apache poslužitelja, definirati upotrebom `ProxyReversePass` direktive. U nastavku je dan primjer korištenja `ProxyPassReverse` direktive, kojom će se implementirati reverzno *proxy* pravilo. Sintaksa korištenja spomenute direktive je sljedeća:

```
ProxyPassReverse lokalni_put interni_posluzitelj
```

pri čemu argumenti direktive imaju sljedeće značenje

- `lokalni_put` – niz s kojim će *reverse proxy* Apache poslužitelj zamijeniti elemente HTTP odgovora koji sadrže informacije o *backend* poslužiteljima;
- `virtualni_poslužitelj` – adresa poslužitelja s kojeg dolazi odgovor.

Reverzna pravila koja bi odgovarala ranije definiranim regularnim pravilima glasila bi:

```
ProxyPassReverse /www_dokumentacija http://www.test.int/dokumentacija
ProxyPassReverse /arhiva http://www.test.int/arhiva
</VirtualHost>
```

nakon čega je zaključena započeta `VirtualHost` definicija.
Na sličan način moguće je definirati i dodatne virtualne poslužitelje koji će proširiti mogućnosti *reverse proxy* arhitekture.

## 6.4. Tijek komunikacije

U ovom poglavlju biti će dan opis tijeka komunikacije prilikom korištenja *reverse proxy* arhitekture. Biti će opisan način na koji Web klijenti te *reverse proxy* i *backend* poslužitelji međusobno razmjenjuju pakete, zajedno s njihovim ispisom.
Web klijent, odnosno korisnik, komunikaciju započinje iniciranjem HTTP konekcije prema *reverse proxy* poslužitelju www.test.com. Kako bi se osiguralo da svi zahtjevi klijenta za pristup www.test.com poslužitelju budu procesirani od strane *reverse proxy* poslužitelja, unutar javnog DNS sustava za domenu test.com mora postojati odgovarajući A zapis (engl. *resource record*) koji će to omogućiti.

**Napomena:** U ovom primjeru neće biti razmatran tijek komunikacije s DNS poslužiteljem te će se podrazumijevati da klijent zna IP adresu koja je putem DNS sustava povezana s imenom www.test.com.

```
GET /dokumentacija/index.php HTTP/1.1
Accept: */*
Accept-Language: hr
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0)
Host: www.test.com
Connection: Keep-Alive
```

Pri primanu zahtjeva klijenta, *reverse proxy* poslužitelj, prema definiranim pravilima prepisivanja (`RewriteRule` direktiva), zahtjeve prosljeđuje internom *backend* poslužitelju www.test.int. Način dolaska do IP adrese internog poslužitelja ovdje neće biti razmatran. Najjednostavniji način je onaj već ranije spomenuti, u kojem se unutar `/etc/hosts` datoteke na *proxy* poslužitelju unese odgovarajući zapis, koji će ime internog poslužitelja povezati s IP adresom.

```
GET /www_dokumentacija/index.php HTTP/1.1
Host: www.test.int
Accept: */*
Accept-Encoding: gzip, deflate
Accept-Language: hr
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0)
X-Forwarded-For: 161.53.x.x (IP adresa Web klijenta)
X-Forwarded-Host: www.test.com
X-Forwarded-Server: www.test.com
Connection: close
```

*Backend* poslužitelj procesira proslijeđeni zahtjev i vraća odgovor *reverse proxy* poslužitelju.

```
HTTP/1.1 200 OK
Date: Fri, 30 May 2003 09:05:07 GMT
Server: Apache-AdvancedExtranetServer/1.3.26 (Mandrake Linux/6mdk)
mod_ssl/2.8.10 OpenSSL/0.9.6g sxnet/1.2.4 PHP/4.2.3
X-Powered-By: PHP/4.2.3
Connection: close
Transfer-Encoding: chunked
Content-Type: text/html; charset=iso-8859-2

<head>
<title>WWW dokumentacija</title>
<meta  http-equiv="Content-Type"  content="text/html;  charset=iso-8859-
2">
</head>
<frameset rows="100,*" frameborder="NO" border="0" framespacing="0">
<frame name="topFrame" scrolling="NO" noresize src=".php" >
  <frame name="mainFrame" src="/">
</frameset>
<noframes>
<body bgcolor="#FFFFFF" text="#000000">
</body>
</noframes>
</html>
```

U zadnjem koraku *reverse proxy* poslužitelj klijentu vraća isti odgovor, čime se dovršava komunikacija.

```
HTTP/1.1 200 OK
Date: Fri, 30 May 2003 09:11:33 GMT
Server:  Apache-AdvancedExtranetServer/1.3.26  (Mandrake  Linux/6mdk)
mod_ssl/2.8.10 OpenSSL/0.9.6g sxnet/1.2.4 PHP/4.2.3
X-Powered-By: PHP/4.2.3
```

```
Content-Type: text/html; charset=iso-8859-2
X-Cache: MISS from www.test.com
Keep-Alive: timeout=15, max=98
Connection: Keep-Alive
Transfer-Encoding: chunked

<head>
<title>WWW dokumentacija</title>
<meta  http-equiv="Content-Type"  content="text/html;  charset=iso-8859-
2">
</head>
<frameset rows="100,*" frameborder="NO" border="0" framespacing="0">
<frame name="topFrame" scrolling="NO" noresize src=".php" >
<frame name="mainFrame" src="/">
</frameset>
<noframes>
<body bgcolor="#FFFFFF" text="#000000">
</body>
</noframes>
</html>
```

## 6.5. Testiranje sustava

Nakon što su podešeni svi parametri uspostavljenog *reverse proxy* sustava, potrebno je testirati njegovu funkcionalnost. U svrhu testiranja moguće je privremeno unutar konfiguracijske datoteke Apache poslužitelja direktivi `RewriteLogLevel` pridijeliti maksimalnu vrijednost 9.
Nakon toga moguće je Web preglednik uputiti na adresu www.test.com te provjeriti da li se Web sadržaj sa *backend* poslužitelja ispravno prikazuje. U nastavku je izdvojen dio sadržaja `/usr/local/apache/logs/public_rewrite_log` datoteke sa Apache *reverse proxy* poslužitelja (www.test.com), u kojoj su zabilježeni podaci o prosljeđivanju zahtjeva na interni *backend* poslužitelj:

```
161.53.x.x - - [29/May/2003:11:00:14 +0200]
[www.test.com/sid#808f468][rid#80cb5a0/initial] (2) init rewrite engine
with requested uri /dokumentacija

161.53.x.x - - [29/May/2003:11:00:14 +0200]
[www.test.com/sid#808f468][rid#80cb5a0/initial] (3) applying pattern
'^/dokumentacija/(.*)' to uri '/dokumentacija'

161.53.x.x - - [29/May/2003:11:00:14 +0200]
[www.test.com/sid#808f468][rid#80cb5a0/initial] (3) applying pattern
'^proxy:.*' to uri '/dokumentacija'

161.53.x.x - - [29/May/2003:11:00:14 +0200]
[www.test.com/sid#808f468][rid#80cb5a0/initial] (3) applying pattern
'^(.*\/perl\/.*)$' to uri '/dokumentacija'

161.53.x.x - - [29/May/2003:11:00:14 +0200]
[www.test.com/sid#808f468][rid#80cb5a0/initial] (3) applying pattern
'^(.*\/cgi-perl\/.*)$' to uri '/dokumentacija'

161.53.x.x - - [29/May/2003:11:00:14 +0200]
[www.test.com/sid#808f468][rid#80cb5a0/initial] (1) pass through
/dokumentacija

161.53.x.x - - [29/May/2003:11:00:44 +0200]
[www.test.com/sid#808f468][rid#80cb5a0/initial] (2) init rewrite engine
with requested uri /dokumentacija/index.php

161.53.x.x - - [29/May/2003:11:00:44 +0200]
[www.test.com/sid#808f468][rid#80cb5a0/initial] (3) applying pattern
'^/dokumentacija/(.*)' to uri '/dokumentacija/index.php'

161.53.x.x - - [29/May/2003:11:00:44 +0200]
[www.test.com/sid#808f468][rid#80cb5a0/initial] (2) rewrite
```

```
/dokumentacija/index.php ->
http://www.test.int/www_dokumentacija/index.php

161.53.x.x - - [29/May/2003:11:00:44 +0200]
[www.test.com/sid#808f468][rid#80cb5a0/initial] (2) forcing proxy-
throughput with http://www.test.int/www_dokumentacija/index.php

161.53.x.x - - [29/May/2003:11:00:44 +0200]
[www.test.com/sid#808f468][rid#80cb5a0/initial] (1) go-ahead with proxy
request proxy:http://www.test.int/www_dokumentacija/index.php [OK]

161.53.x.x - - [29/May/2003:11:00:44 +0200]
[www.test.com/sid#808f468][rid#80cb5a0/initial] (2) init rewrite engine
with requested uri /

161.53.x.x - - [29/May/2003:11:00:44 +0200]
[www.test.com/sid#808f468][rid#80cb5a0/initial] (3) applying pattern
'^/dokumentacija/(.*)' to uri '/'.

.

.
```

U gornjem primjeru žutom bojom označeni su zapisi koji potvrđuju definirano prosljeđivanje zahtjeva. Na sličan način moguće je u log datotekama *backend* poslužitelja provjeriti adresu s koje pristižu HTTP upiti.

```
192.168.1.2 - - [30/May/2003:10:14:09 +0200] "GET /www_dokumentacija/
HTTP/1.1" 200 402 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT
5.0)"

192.168.1.2 - - [30/May/2003:10:14:09 +0200] "GET
/www_dokumentacija/index.php HTTP/1.1" 404 329
"http://www.test.com/dokumentacija/" "Mozilla/4.0 (compatible; MSIE
6.0; Windows NT 5.0)"

192.168.1.2 - - [30/May/2003:10:14:13 +0200] "GET
/www_dokumentacija/index.php HTTP/1.1" 404 329
"http://www.test.com/dokumentacija/" "Mozilla/4.0 (compatible; MSIE
6.0; Windows NT 5.0)"
```

Adresa 192.168.1.2 u ovom slučaju predstavlja adresu *reverse proxy* Apache poslužitelja, kojem je putem DNS sustava pridjeljeno ime www.test.com.

# 7. Zaključak

U dokumentu je opisana tehnologija *reverse proxy* poslužitelja, zajedno s njezinim osnovnim karakteristikama, prednostima, nedostacima, mogućnostima upotrebe i sl. U zadnjem dijelu dan je primjer implementacije *reverse proxy* sustava pomoću Apache Web poslužitelja, s osnovnim smjernicama kojih se treba pridržavati prilikom implementacije sustava ovog tipa. Također je analiziran i tijek komunikacija između pojedinih komponenti sustava te ispis sadržaja pojedinih paketa čime se dodatno demonstrirao način rada sustava.