CARNet

HRVATSKA AKADEMSKA I ISTRAŽIVAČKA MREŽA
CROATIAN ACADEMIC AND RESEARCH NETWORK

Proxy poslužitelji

NCERT-PUBDOC-2010-08-309

Nacionalni CERT

u suradnji s

LSS

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

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

Nacionalni 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

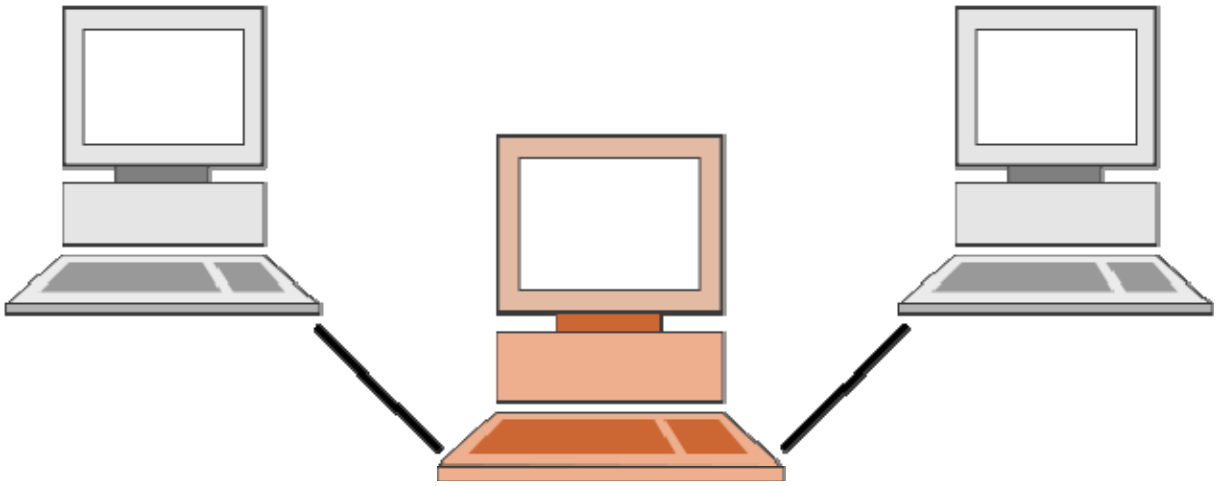
1. Uvod

U današnje vrijeme, očuvanje sigurnosti i anonimnosti u komunikaciji putem Interneta je nešto čemu svi teže. Na taj način se izbjegavaju napadi na računala korisnika koji mogu proizvesti nepotrebne poteškoće. Jednom kada napadač otkrije najvažnije podatke o računalu poput njegove IP adrese, iskorištavajući sigurnosne propuste u operacijskom sustavu i/ili korištenim programskim paketima, na njega može postaviti zloćudni program ili nanijeti drugu štetu korisniku. *Proxy* poslužitelji mogu „sakriti" korisnika i time spriječiti većinu napada. Budući da djeluju kao posrednici u komunikaciji između klijenta i poslužitelja, oni nadgledaju sve zahtjeve koji kroz njih prolaze i imaju mogućnost njihovog mijenjanja kako bi se postigla određena razina sigurnosti i anonimnosti. *Proxy* poslužitelji su u osnovi vrlo jednostavni za korištenje, a mogu se koristiti u različite svrhe.

U ovom dokumentu će biti objašnjeni svi razlozi korištenja *proxy* poslužitelja. Jedan razlog je već spomenut - očuvanje anonimnosti korisnika, ali postoje još mnogi drugi. Budući da kroz *proxy* poslužitelj prolaze klijentski zahtjevi, *proxy* poslužitelj može bilježiti sve zahtjeve koji prolaze kroz njega i tako nadgledati cijeli promet krajnjih korisnika, ali moguće ga je koristiti i za filtriranje klijentskih zahtjeva. U dokumentu je objašnjen način na koji *proxy* poslužitelji rade i koliko dobro taj posao mogu obavljati.

2. Namjena *proxy* poslužitelja

Proxy je računalo koje djeluje kao posrednik između klijenta i poslužitelja. Klijent se povezuje na *proxy* poslužitelj tražeći neku uslugu (datoteku, *web* stranicu ili nešto drugo) od drugog poslužitelja. Klijentski zahtjev se, nakon obrade unutar *proxy* poslužitelja, prosljeđuje do željenog poslužitelja (u izvornom ili nešto izmijenjenom obliku) ili odbacuje ako nije zadovoljio neki zadanih uvjeta postavljenih na *proxy* poslužitelju. Na slici 1 *proxy* poslužitelj je simbolički prikazan sa crvenim računalom koji se nalazi između druga dva računala – klijenta i poslužitelja. Svi zahtjevi s klijentskog računala prolaze kroz *proxy* poslužitelj prije nego dođu do odredišnog poslužitelja. Poslužiteljev odgovor na klijentski zahtjev vraća se istim putem.



Slika 1.* Proxy *poslužitelj
Izvor: Wikipedia

Proxy poslužitelji se koriste u razne svrhe kao što su:

- anonimnost klijentskih računala tj. krajnjih korisnika,
- ubrzanje pristupa resursima upotrebom metode privremene pohrane (eng. *caching*)
- zabrana pristupa određenim *web* stranicama,
- zabrana pristupa *web* stranicama s određenim ključnim riječima,
- zabrana određenih protokola,
- zabrana pristupa određenim priključcima (eng. *ports*),
- zabrana pristupa određenih korisnika *proxy* poslužitelju,
- praćenje korisnikovih zahtjeva,
- zaobilaženje zabrana pristupa,
- pretraživanje sadržaja koji se prenosi ili
- uklanjanje dijelova *web* stranice poput reklama.

Najčešća upotreba *proxy* poslužitelja je ubrzanje mrežnog prometa upotrebom *caching*-a i filtriranje zahtjeva. *Caching* je postupak kojim se često dohvaćan sadržaj sprema u memoriji *proxy* poslužitelja kako se ne bi morao dohvaćati s udaljenog poslužitelja svaki put kad klijent podnese zahtjev. Prvi *proxy* poslužitelji su bili upravo ovog oblika i koristili su se u velikim tvrtkama kako bi se smanjili troškovi Internet prometa. S njima su se mogle dopustiti ili zabraniti određene *web* stranice, ali i nadgledati cjelokupni promet zaposlenika te tvrtke.

Danas se *proxy* poslužitelji često koriste za zaobilaženje zabrana koje su postavili drugi *proxy* poslužitelji. Za to se koriste razne tehnike poput tuneliranja, gdje se *proxy* poslužitelj koristi kako bi korisnik došao do poslužitelja kojem inače ne može pristupiti.

Upotrebom *proxy* poslužitelja korisnik može ostvariti anonimnost, jer svi zahtjevi do odredišnog poslužitelja dolaze od *proxy* poslužitelja (koji se predstavlja kao klijent). Korisnikovi podaci ostaju skriveni, a on je došao do sadržaja koji je tražio.

3. Vrste *proxy* poslužitelja

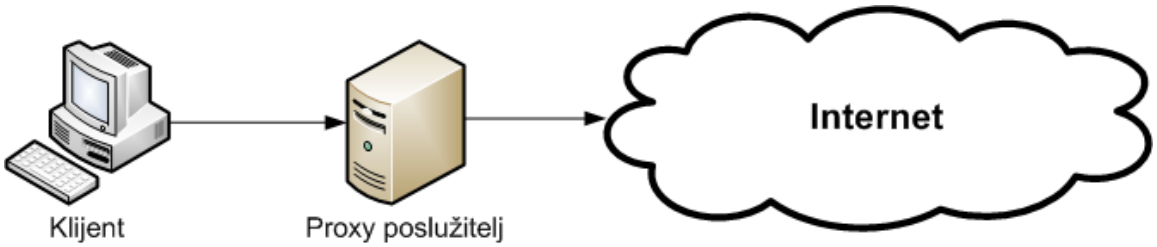
Zbog toga što je *proxy* poslužitelj jedno računalo kojeg se može programirati da radi više različitih poslova, podjela *proxy* poslužitelja može se napraviti na više načina. Svaka podjela nije jednoznačna jer se svaki poslužitelj može programirati da radi posao zbog kojeg bi se mogao svrstati u drugu skupinu. Danas se *proxy* poslužitelji najčešće koriste u dva oblika. Jedan čine *proxy* poslužitelji kroz koje zahtjevi klijenata uvijek prolaze jer je mreža tako podešena. Takvi *proxy* poslužitelji nalaze se u tvrtkama, školama ili drugim javnim mjestima te nadgledaju izlazni Internet promet. Mogu imati nekoliko funkcija poput ubrzavanja i filtriranja prometa ili se jednostavno koriste za postavljanje zabrana pristupa određenim *web* stranicama. Drugi oblik čine *proxy* poslužitelji na koje se klijent svjesno spaja kako bi pomoću njega zaobišao neku zabranu koju je postavio drugi *proxy* poslužitelj ili odredišni poslužitelj (npr. dozvola pristupa samo s određenih IP adresa). Ali to nisu jedini razlozi upotrebe *proxy* poslužitelja. Ostali oblici *proxy* poslužitelja biti će objašnjeni u nastavku.

Proxy poslužitelji se, također, mogu razlikovati po mjestima na koja se postavljaju. Oni mogu biti bliže klijentima, bliže poslužiteljima ili negdje između klijenta i poslužitelja, odnosno negdje na Internetu. Ako se postavlja bliže klijentu, odnosno poslužitelju, onda svi zahtjevi uvijek prolaze kroz *proxy* poslužitelj prije nego dođu do svog odredišta, često bez da to klijenti znaju. Ako se koristi *proxy* poslužitelj koji je smješten negdje na Internetu, klijent svjesno svoje zahtjeve preusmjerava preko *proxy* poslužitelja kada mu to zatreba.

3.1. Caching proxy poslužitelj

Caching proxy poslužitelj se nalazi blizu klijenata i svi klijentski zahtjevi prolaze prvo kroz njega, a tek zatim dolaze do svog odredišta. Ovakav *proxy* poslužitelj služi za ubrzavanje prometa jer čuva podatke koji se često dohvaćaju kako se oni ne bi morali svaki puta iznova dohvaćati s udaljenog poslužitelja. Sadržaj dohvaćen iz prethodnih zahtjeva klijenata čuva se u memoriji *proxy* poslužitelja određeno vrijeme (jer će vjerojatno i drugi klijenti te tvrtke trebati dohvatiti isti sadržaj). To može biti sličica loga tvrtke koja se nalazi na svakoj *web* stranici te tvrtke, dokument koji trebaju pročitati svi zaposlenici tvrtke ili neki drugi sadržaj koji je takve veličine da bi njegov učestali dohvat zahtijevao veliki dio prometa te tvrtke. Ova funkcionalost *proxy*-ja znatno povećava brzinu, kvalitetu veze i smanjuje količinu podataka koji se trebaju dohvaćati, što u konačnici može rezultirati materijalnom uštedom. Većina ISP-ova (eng. *Internet Service Provider*) i velikih tvrtki imaju ovakve *proxy* poslužitelje. Oni su bili i prva vrsta *proxy* poslužitelja.

3.2. Web proxy poslužitelj



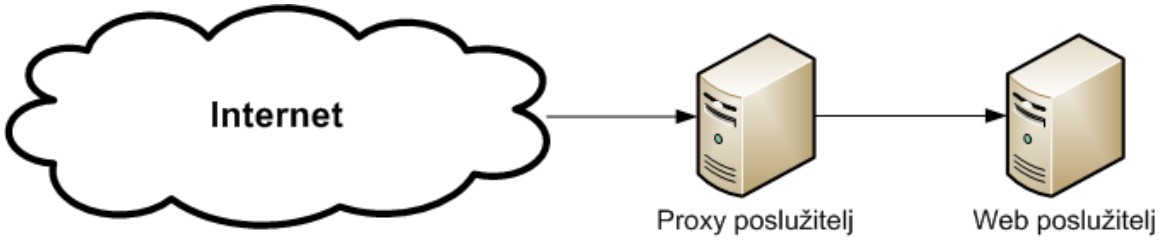
Slika 2. Dijagram* web proxy *poslužitelja

Web proxy poslužitelj se postavlja na isto mjesto kao i *caching proxy* poslužitelj te ima sličnu zadaću, ali je više orijentiran na *web* promet na Internetu. Slika 2 prikazuje na koje mjesto se postavlja proxy poslužitelj ove vrste. Svi zahtjevi s klijenata uvijek prolaze prvo kroz proxy poslužitelj prije nego što dođu do odredišnog poslužitelja. *Web proxy* poslužitelj čuva sadržaj ili dijelove sadržaja prethodno dohvaćenih *web* stranica s Interneta kako bi se ubrzao promet, zbog čega bi se mogao svrstati u skupinu *caching proxy* poslužitelja, ali sadrži i dodatnu mogućnost. On omogućuje filtriranje zahtjeva klijenata na temelju URL-a, DNS crne liste ili sadržaja. Ovakvi *proxy* poslužitelji se često koriste u tvrtkama, školama, knjižnicama i drugim ustanovama gdje je potrebno filtriranje web sadržaja. Često korisnici mogu zaobići ovakve *proxy* poslužitelje upotrebom drugih *proxy* poslužitelja metodom koja se zove tuneliranje a koja će biti objašnjena u poglavljima koja slijede.

Neki *web proxy* poslužitelji preuređuju *web* stranice kako bi one bile prikladne za prikaz na mobilnim uređajima koji imaju mali ekran. Prije slanja *web* stranice klijentu, slike se sažimaju i smanjuje im se

dimenzije, izbacuju se dijelovi stranice koji ne mogu biti prikazani na mobilnim uređajima (poput flash animacija), a tekst se oblikuje tako da nije potrebno horizontalno pomicanje da se pročita tekst, već samo vertikalno. Ovo rezultira znatno ugodnijim čitanjem *web* stranice na mobilnom uređaju, a i smanjuje se količina podataka koja se dohvaća te time štedi novac.

3.3. Reverzni proxy poslužitelj

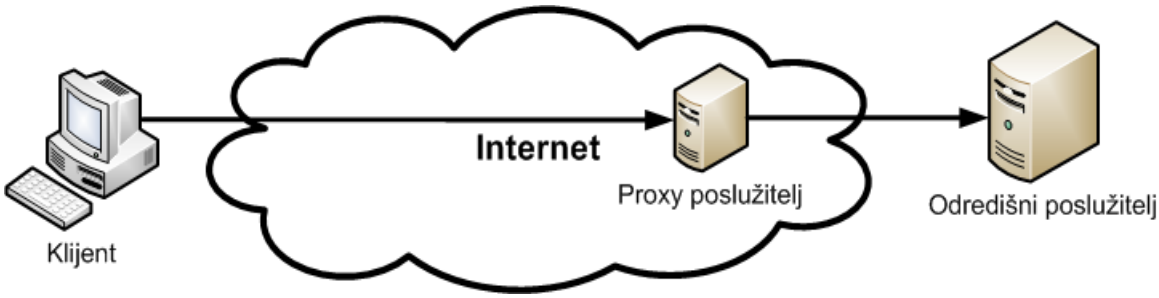


Slika 3. Dijagram reverznog* proxy *poslužitelja

Reverzni *proxy* poslužitelj se postavlja ispred *web* poslužitelja (slika 3), a sav promet s Interneta prolazi prvo kroz *proxy* poslužitelj koji obrađuje zahtjeve prije nego oni dođu do *web* poslužitelja. *Proxy* poslužitelj zahtjeve može odbaciti ili proslijediti u izvornom ili izmijenjenom obliku. Služi kako bi se rasteretili poslužitelji koji stoje „iza" njega jer može odbacivati zahtjeve koji ne udovoljavaju kriterijima i preusmjeravati zahtjeve na druge poslužitelje koji u tom trenutku obrađuju manje zahtjeva. Reverzni *proxy* poslužitelj štiti poslužitelje iza sebe jer klijent vidi samo *proxy* poslužitelj i ne može izravno pristupiti krajnjim aplikacijskim (npr. web) poslužiteljima i time napraviti štetu. Na njega se može dograditi i funkcionalnost vatrozida koja će omogućiti aktivni nadzor prometa, te na taj način isključiti mogućnost većeg broja napada. Također, *proxy* poslužitelj može obavljati SSL enkripciju umjesto npr. *web* poslužitelja. Tada se na *proxy* poslužitelj postavlja sklopovski ili programski ubrzivač enkripcije. Prednost je što je potrebno imati samo jedan certifikat za cijelu skupinu poslužitelja (umjesto zasebnih certifikata za svaki poslužitelj pojedinačno). Nedostatak je što svi poslužitelji iza *proxy* poslužitelja moraju dijeliti isto DNS ime ili IP adresu SSL veze.

Ostali zadaci koje može izvoditi ovakav *proxy* poslužitelj su isti kao kod *web proxy* poslužitelja. On može filtrirati zahtjeve gledajući IP adresu pošiljatelja zahtjeva, raditi *caching* kako bi ubrzao promet ili komprimirati stranice kako bi se one prilagodile mobilnim uređajima.

3.4. *Anonimni* proxy *poslužitelji*



Slika 4. Dijagram anonimnog* proxy *poslužitelja

Anonimni *proxy* poslužitelji služe za skrivanje identiteta klijenata koji se preko njih povezuju s drugim poslužiteljima. Za to se najčešće koriste otvoreni *proxy* poslužitelji (eng. *open proxy*) poput CGI *proxy* poslužitelja kojima može pristupiti bilo koji Internet korisnik preko *web* stranice *proxy* poslužitelja. Ovakvi *proxy* poslužitelji su zanimljivi napadačima i ljudima kojima tijela kaznenog progona prate pristup Internet stranicama, ali i osobama koje žele zaštititi svoju privatnost dok pretražuju Internet (kako bi što manje bili izloženi napadima). Svaki posjet korisnika nekoj *web* stranici se bilježi, a te informacije se kasnije mogu koristiti u marketingu ili za utvrđivanje odgovornosti. Poznavanjem IP adrese i operacijskog sustava računala mogu se iskoristiti sigurnosne rupe u operacijskom sustavu računala i postaviti zlćudni programi. Anonimni *proxy* poslužitelji skrivaju IP adresu korisnika i tako ga štite od

mogućeg napada. Razni anonimni *proxy* poslužitelji skrivaju IP adresu s više ili manje uspjeha. Naime, veliki broj anonimnih *proxy* poslužitelja ipak šalje korisnikovu IP adresu u nekom obliku. Oni prosljeđuju pakete s HTTP naredbama HTTP_VIA, HTTP_X_FORWARDED_FOR ili HTTP_FORWARDED koji mogu otkriti izvornu IP adresu klijenta. Pravi anonimni *proxy* poslužitelji, koji se još nazivaju i elitni *proxy* poslužitelji, uz skrivanje IP adrese, od poslužitelja skrivaju činjenicu da zahtjev dolazi od jednog *proxy* poslužitelja. Ciljni poslužitelj dobiva zahtjev od *proxy* poslužitelja kojeg smatra klijentom i zbog toga ne zna ništa o korisnikovom računalu (ali zato *proxy* poslužitelj zna sve korisnikove podatke). Zbog toga korisnik nikada ne može ostati potpuno anoniman jer se njegovi podaci uvijek mogu naći u *proxy* poslužitelju.

Budući da su napadi preko anonimnih *proxy* poslužitelja vrlo česti, kako bi se spriječio pristup *web* stranicama preko ovakvih *proxy* poslužitelja, administratori su razvili niz načina kojima provjeravaju dolazi li zahtjev preko anonimnog *proxy* poslužitelja. Najčešće se provjeravaju IP adrese često korištenih anonimnih *proxy* poslužitelja.

4. Praktična primjena

Na početku ovog dokumenta spomenuto je kako se *proxy* poslužitelji pojavljuju u dva glavna oblika. Jedan oblik čine oni *proxy* poslužitelji kroz koje uvijek prolaze zahtjevi klijenata koji se nalaze u istoj mreži, često bez da klijenti to znaju. Takvi *proxy* poslužitelji poznati su kao klasični *proxy* poslužitelji. Drugi oblik čine *proxy* poslužitelji na koje klijenti svjesno preusmjeravaju svoje zahtjeve. Oni se koriste za metodu koja se zove tuneliranje, a služi za zaobilaženje zabrana pristupa određenim *web* stranicama.

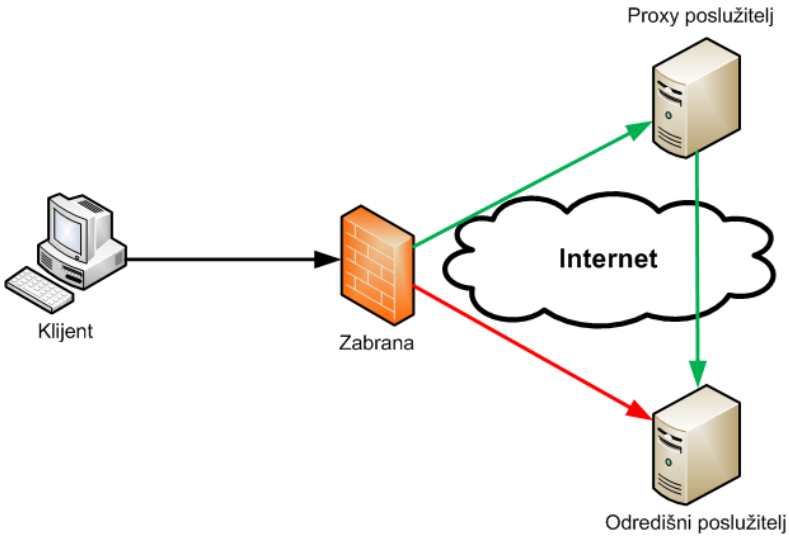
4.1. Klasični proxy poslužitelji

Klasični *proxy* poslužitelji su najčešći korišteni u praksi. Koriste se, kako u velikim tvrtkama koje žele uštedjeti i ubrzati svoj Internet promet, tako i na osobnim računalima korisnika. Brojne su besplatne implementacije koje ostvaruju ubrzanje spremanjem često dohvaćanog sadržaja u lokalnu memoriju *proxy* poslužitelja metodom zvanom *caching*. Neke besplatne implementacije su opisane u poglavlju 4. S vremenom su ovi *proxy* poslužitelji dobili mogućnost filtriranja zahtjeva i postavljanja zabrana pristupa nekim *web* stranicama. Oni se najčešće koriste u tvrtkama, knjižnicama i ostalim javnim ustanovama. Zabrane implementirane ovom vrstom proxy poslužitelja predstavljaju problem nekim korisnicima pa je nastao drugi oblik *proxy* poslužitelja koji zaobilazi te zabrane metodom tuneliranja.

4.2. Tuneliranje

Tuneliranje je metoda za zaobilaženje zabrana pristupa određenim *web* stranicama ili drugim servisima. U ovoj metodi koristi se *proxy* poslužitelj kako bi se dobio pristup *web* stranicama koje su inače nedostupne. *Proxy* poslužitelju za tuneliranje se pristupa preko obrasca na *web* stranici, posebnog programa ili Internet preglednika koji je podešen za komunikaciju s *proxy* poslužiteljem. Svaki od ovih načina bit će objašnjen u nastavku. Međutim, u svim slučajevima cilj je klijentski zahtjev tunelirati do *proxy* poslužitelja koji može dobiti sadržaj koji je klijentu zabranjen. *Proxy* poslužitelj prosljeđuje zahtjev do krajnjeg poslužitelja i vraća odgovor tuneliranjem do klijenta. Na taj način je klijent došao do informacija koje mu inače nisu bile dostupne.

Na slici 5 prikazan je postupak tuneliranja. Klijent zbog zabrane ne može pristupiti odredišnom poslužitelju (crvena linija). Međutim, put do *proxy* poslužitelja nije zabranjen, te se klijentski zahtjevi preusmjeravaju preko *proxy* poslužitelja. Odgovor odredišnog poslužitelja se vraća istim putem, preko *proxy* poslužitelja. Put koji povezuje klijenta i odredišni poslužitelj preko *proxy* poslužitelja postao je tunel kojim se zaobilazi postavljena zabrana.



Slika 5. Dijagram tuneliranja

Metodu tuneliranja često upotrebljavaju korisnici kojima je vlada (ili neko drugo zakonodavno tijelo) zabranila pristup određenim stranicama sa sadržajem kojeg smatraju nepodobnim. Takvim korisnicima je tuneliranje najbolje rješenje ako izvan države imaju dostupno računalo preko kojeg mogu slati svoje zahtjeve. Tada se uspostavi tunel između korisnika i računala koje ima pristup zabranjenom sadržaju, i zatim dalje svi zahtjevi prolaze kroz uspostavljeni tunel. To računalo tada služi kao *proxy* poslužitelj.

Ako korisnik nema tu mogućnost, postoji mogućnost plaćanja komercijalnih sustava za tuneliranje ili upotreba besplatnih *proxy* poslužitelja. Ovakav oblik tuneliranja ne štiti korisnike u istoj mjeri kao prethodno opisani postupak jer ne pružaju svi *proxy* poslužitelji jednaku anonimnost, a nekada je čak potrebna i registracija za upotrebu *proxy* poslužitelja. Tada se čuvaju korisnički podaci preko kojih se može otkriti korisnika koji radi suprotno zabrani. Može se dogoditi da je ovakve stranice postavila državna služba s namjerom otkrivanja ljudi koji pokušavaju doći do zabranjenog sadržaja. Na kraju, država može zabraniti pristup stranicama koje pružaju uslugu tuneliranja.

Drugi tip korisnika kojima je tuneliranje zanimljivo rješenje su oni kojima je zabranjen pristup medijskom sadržaju zato što ne žive u određenoj državi. Često je video sadržaj na nekoj *web* stranici dostupan samo državljanima npr. Sjedinjenih Američkih Država ili Velike Britanije. Državljani ostalih zemalja mogu vidjeti samo poruku da je taj video sadržaj za njih nedostupan jer se nalaze u 'krivoj' državi. Tuneliranjem, i ti korisnici dobivaju pristup i mogu pogledati video inače rezerviran za državljane Sjedinjenih Američkih Država. Korisnici neće imati problema sa zakonom ako ih se otkrije, pa ne mare za anonimnost. Jedini nedostatak je duže vrijeme potrebno za dohvat sadržaja zbog preusmjeravanja.

Danas postoji niz servisa za tuneliranje, neki su besplatni, neki nisu, neki zahtijevaju instalaciju programa na korisnikovo računalo, a neki su dostupni preko *web* stranice. Među najpoznatijim besplatnim programima su Ultrareach, Freegate i Tor koji nakon instalacije i nešto dodatnog konfiguriranja Internet preglednika uspješno zaobilaze zabrane. Više podataka o navedenim programima može se naći na sljedećim *web* stranicama:

http://www.ultrareach.com/

http://www.dit-inc.us

Anonymizer je program koji je nešto jednostavniji od prethodna dva, ali nije besplatan. Ipak, zanimljiv je neiskusnijim korisnicima koji su spremni platiti takvu uslugu. Ostali programi koji se mogu koristiti za tuneliranje mogu se naći na sljedećoj *web* stranici:

http://www.dmoz.org/Computers/Internet/*Proxy*ing_and_Filtering/Products_and_Tools/Software/

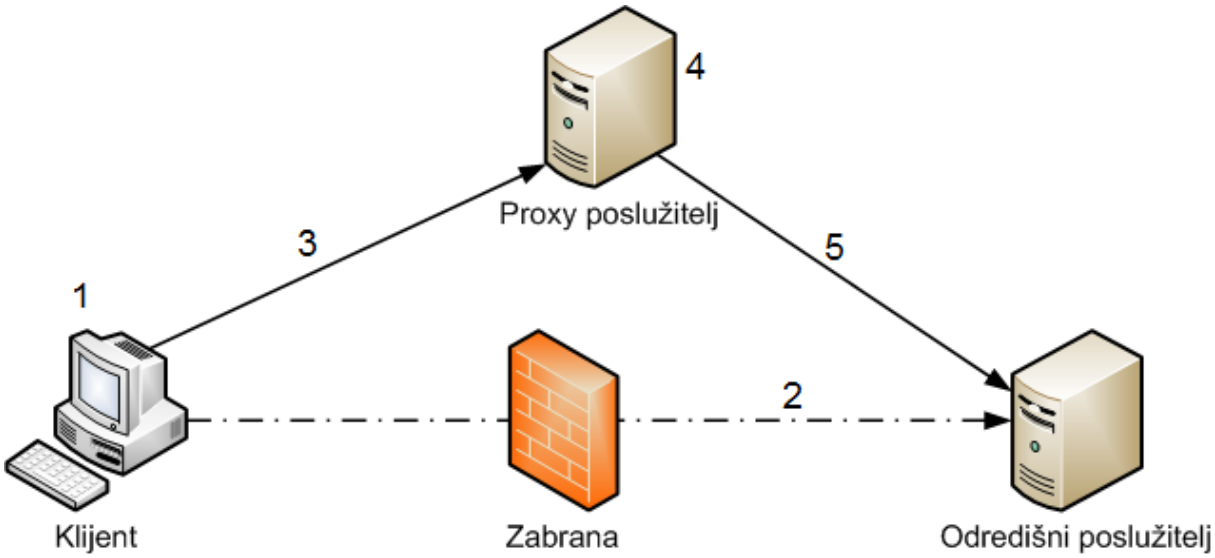
Ovakvi programi daju rješenje korisnicima koji se koriste svojim osobnim računalima za spajanje na Internet. Ako im je dostupno javno računalo poput računala u knjižnici, školi, na poslu ili jednostavno ne žele instalirati dodatne programe na svoje računalo, jednostavnije im je koristiti *proxy* poslužitelje kojima se pristupa preko *web* stranice. Takvih poslužitelja ima jako puno, ali nisu svi sigurni i pouzdani, a mnogima je pristup i zabranjen. Za korištenje su najjednostavniji CGI *proxy* poslužitelji za koje nije potrebno podešavanje Internet preglednika, nego je samo na *web* stranici *proxy* poslužitelja potrebno upisati URL stranice do koje se želi doći. Popis nekih CGI *proxy* poslužitelja može se naći na sljedećoj *web* stranici:

http://www.dmoz.org/Computers/Internet/*Proxy*ing_and_Filtering/Hosted_*Proxy*_Services/Free/CGI_*Proxy*/

Na Internetu se može naći veliki broj *proxy* poslužitelja, ali većinom nisu potpuno anonimni. U nastavku će biti objašnjen postupak tuneliranja kroz tri vrste *proxy* poslužitelja.

4.2.1. HTTP *proxy* poslužitelj

HTTP *proxy* poslužitelji su najčešće korišteni oblik *proxy* poslužitelja za tuneliranje. Kao što im ime govori, podržavaju samo HTTP protokol. Rijetki podržavaju i FTP protokol. Korisnici ih obično upotrebljavaju kako bi došli do *web* stranice koja im je inače zabranjena. Postupak slanja korisnikovog zahtjeva i njegovog preusmjeravanja preko *proxy* poslužitelja prikazan je na sljedećoj slici.



Slika 6. Dijagram preusmjeravanja preko HTTP* proxy *poslužitelja

Postupak je sljedeći:

1. Korisnik podešava svoj Internet preglednik tako da on koristi HTTP *proxy* poslužitelj.
2. Korisnik šalje zahtjev prema odredišnom poslužitelju kao da nema zabrane.
3. Korisnikov zahtjev dolazi do *proxy* poslužitelja.
4. *Proxy* poslužitelj čita korisnikov zahtjev kako bi saznao kamo zahtjev treba ići.
5. *Proxy* poslužitelj šalje korisnikov zahtjev do odredišnog poslužitelja.

Poslužiteljev odgovor vraća se istim putem kojim je došao klijentski zahtjev, dakle preko *proxy* poslužitelja. Odredišni poslužitelj odgovor šalje *proxy* poslužitelju, a on ga dalje prosljeđuje do klijenta. Sa stajališta klijenta, odgovor je došao izravno s odredišnog poslužitelja, a ne preko *proxy* poslužitelja.

Svi Internet preglednici su sposobni za rad s HTTP *proxy* poslužiteljima, potrebno je samo podesiti preglednik, dakle upisati adresu i priključnicu *proxy* poslužitelja. Nakon toga korisnik se njime koristi kao da se ništa nije promijenilo, iako će svi zahtjevi prolazi kroz HTTP *proxy* poslužitelj.

HTTP *proxy* poslužitelji imaju nekoliko razina anonimnosti:

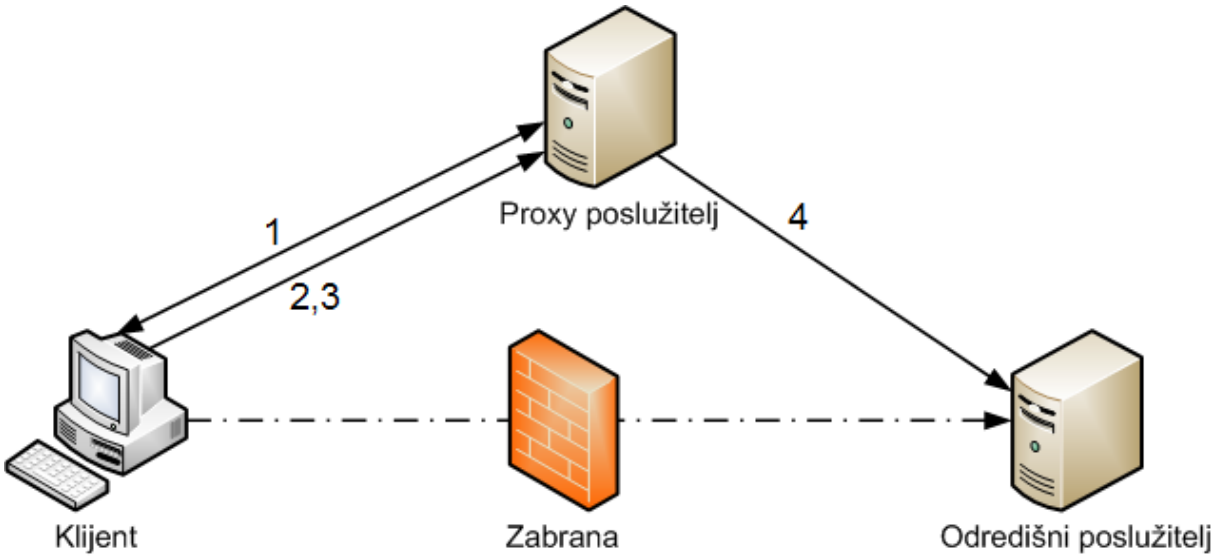
- **Transparentni** – djeluju samo kao posrednici, tj. ne skrivaju klijentsku IP adresu.
- **Anonimni** – ne prenose korisnikovu IP adresu, ali daju do znanja da je korišten *proxy* poslužitelj.
- **Distorzijski** – mijenjaju IP adresu korisnika koju šalju do poslužitelja tako da poslužitelj misli da zna s kojim klijentom radi.
- **Elitni** - ne prenose korisnikovu IP adresu i skrivaju činjenicu da se koristi *proxy* poslužitelj.

4.2.2. SOCKS *proxy* poslužitelj

SOCKS je protokol koji omogućuju preusmjeravanje paketa između klijenta i poslužitelja preko *proxy* poslužitelja. Funkcionira na u petom (sjedničkom) sloju OSI modela. Za SOCKS protokol je rezervirana priključnica (eng. *port*) 1080. Za razliku od HTTP *proxy* poslužitelja, SOCKS *proxy* poslužitelj može raditi s više protokola. Razlog je što on ne čita korisnikov zahtjev kako bi saznao kamo zahtjev treba ići, već ga samo prosljeđuje kamo mu korisnik kaže da treba ići. Zbog toga se može koristiti sa sljedećim protokolima:

- HTTP,
- FTP, SMTP,
- POP3 i
- NNTP.

Postupak slanja korisnikovog zahtjeva i njegovog preusmjeravanja preko *proxy* poslužitelja prikazan je na sljedećoj slici.



Slika 7. Dijagram preusmjeravanja preko SOCKS* proxy *poslužitelja

Postupak je sljedeći:

1. Između klijenta i SOCKS *proxy* poslužitelja uspostavi se TCP veza.
2. Klijent šalje SOCKS *proxy* poslužitelju adresu poslužitelja s kojim želi komunicirati.
3. Klijent šalje prema SOCKS *proxy* poslužitelju sadržaj koji želi da odredišni poslužitelj dobije.
4. SOCKS *proxy* poslužitelj stvara zahtjev i šalje klijentski sadržaj do poslužitelja.

SOCKS *proxy* poslužitelj preuzima sadržaj koji je dobio od klijenta i šalje ga do poslužitelja ponašajući se kao da je on klijent. Odredišni poslužitelj ne može saznati tko je pravi tvorac sadržaja kojeg je primio, osim ako to nije navedeno u sadržaju. SOCKS poslužitelj može prosljeđivati

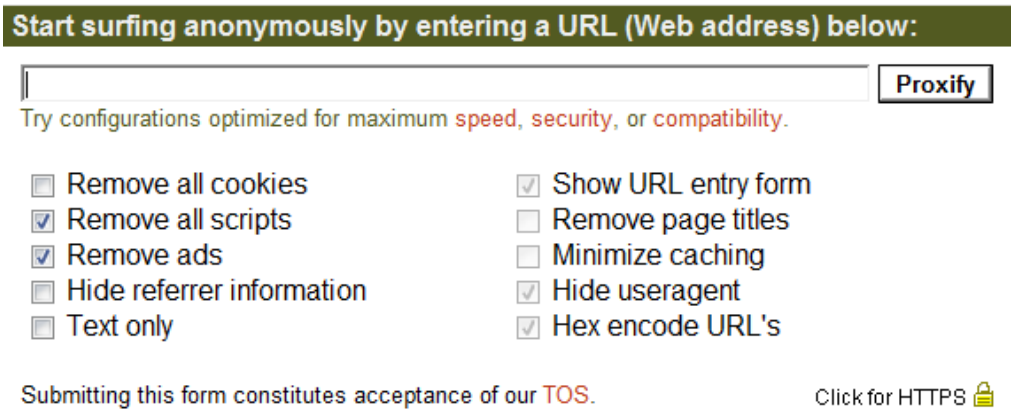
klijentski sadržaj, ali ga je moguće konfigurirati i da ga mijenja ako je to potrebno kako bi se postigla bolja anonimnost klijenta. Odredišni poslužitelj svoj odgovor šalje *proxy* poslužitelju za kojeg misli da je klijent, a *proxy* poslužitelj taj odgovor prenosi do klijenta preko već uspostavljene TCP veze.

Za korištenje HTTP *proxy* poslužitelja potrebno je samo promijeniti nekoliko postavki u Internet pregledniku. Sa SOCKS *proxy* poslužiteljima stvar je nešto složenija. Većina Internet preglednika ne zna raditi s njima, pa je potrebno instalirati dodatni program. Zbog toga je upotreba SOCKS *proxy* poslužitelja nešto složenija od HTTP *proxy* poslužitelja, ali je zato moguće dobiti veću razinu anonimnosti.

Jedan od poznatijih programa za rad sa SOCKS *proxy* poslužiteljima je SocksCap. Njime je moguće koristiti SOCKS *proxy* poslužitelj s bilo kojim programom koji koristi TCP protokol, ne nužno samo s Internet preglednikom. Ako je dostupan *proxy* poslužitelj koji može raditi s novijom inačicom SOCKS protokola (SOCKS 5), onda se preko njega mogu preusmjeravati i UDP paketi.

4.2.3. CGI *proxy* poslužitelj

CGI *proxy* poslužitelji su najjednostavniji za korištenje. Za njihovu upotrebu nije potrebna konfiguracija Internet preglednika niti instalacija dodatnog programa na računalo. *Proxy* poslužitelju se pristupa preko *web* stranice gdje se u posebno polje upisuje URL *web* stranice koju se želi pregledati. Korisnik će potom moći pregledati *web* stranicu do koje inače nije mogao doći. Na slici 8. je prikazan dio *web* stranice poznatog CGI *proxy* poslužitelja imena Proxify. U odgovarajuće polje se upiše URL stranice koja se želi posjetiti, a dodatno se mogu odabrati neke funkcije poput izbacivanja reklama s *web* stranice, isključivanje upotrebe kolačića, raznih skripti itd.



Slika 8. Obrazac za korištenje CGI* proxy *poslužitelja
Izvor: Proxify

U pregledniku će adresa stranice koju korisnik pregledava biti drugačija nego što bi to inače bila. Npr. ako je adresa CGI *proxy* poslužitelja i željene stranice:

```
www.cgi-proxy.com
www.zeljena-stranica.com
```

u adresnoj traci preglednika će stajati adresa slična sljedećoj:

```
http://www.cgi-proxy.com/http/www.zeljena-stranica.com
```

Često je s ovakvim *proxy* poslužiteljima moguće zabraniti izvođenje nekih dijelova stranice poput dijelova napravljenih u JavaScript-u ili zabraniti kolačiće (eng. *cookies*) kako bi se osigurala bolja klijentska anonimnost. Međutim, isključivanje JavaScript-a i VBScript-a može rezultirati netočnim ili nepotpunim prikazivanjem *web* stranice.

Ulančavanje CGI *proxy* poslužitelja, kako bi se povećala anonimnost, je izuzetno jednostavno. Svodi se na upisivanje URL-a jednog CGI *proxy* poslužitelja u drugi i tako nekoliko puta. Na taj način je

moguće bolje sakriti izvornu IP adresu korisnika koji pokušava zaobići zabranu. Također, od administratora se može sakriti pravo ime stranice koja se želi pregledati, pa će adresa iz prethodnog primjera izgledati ovako:

```
http://www.cgi-proxy.com/bcd104dflsjuywe34sdfispd345klksfdsl
```

Nedostatak je što nisu svi CGI *proxy* poslužitelji jednako dobri. Neki uopće ne skrivaju korisnikovu IP adresu, iako to tako tvrde. Za korištenje nekih potrebna je registracija kojom korisnik ostavlja upravo one podatke o sebi koje želi sakriti upotrebom *proxy* poslužitelja. Neki CGI *proxy* poslužitelji su mamci kako bi se pronašli korisnici koji zaobilaze zabrane, kao što je to slučaj u državama u kojima se provodi Internet cenzura. U tim državama zabranjen je pristup *web* stranicama CGI *proxy* poslužitelja, pa su korisnici prisiljeni tražiti nove koje država još nije otkrila i zabranila.

4.2.4. Pregled *proxy* poslužitelja za tuneliranje

U nastavku je tablica koja prikazuje usporedbu spomenutih *proxy* poslužitelja.

Svojstvo	HTTP	SOCKS	CGI
Podržani protokoli	HTTP, rijetko FTP	HTTP, FTP, POP3, SMTP	HTTP, rijetko FTP
Ulančavanje *proxy* poslužitelja	ponekad, složeno	da, složeno	da, jednostavno
Potreban softver	Internet preglednik	zaseban program	Internet preglednik
Lakoća korištenja	potrebno je znati postaviti *proxy* u Internet pregledniku	potrebno je znati postaviti program za rad sa SOCKS *proxy* poslužiteljem	potrebno je osnovno poznavanje Interneta

Tablica 1. Pregled proxy poslužitelja za tuneliranje

5. Besplatne implementacije

Postoje brojna programska rješenja koja implementiraju funkcije *proxy* poslužitelja. Neka od njih su Polipo, Ziproxy i Squid. Od spomenutih rješenja Squid je daleko najrašireniji i najmoćniji. U nastavku je tablica s pregledom svojstava navedenih besplatnih *proxy* poslužitelja.

	Squid	Polipo	*Ziproxy*
Operacijski sustav	gotovo svi	gotovo svi	Unix
Podržani protokoli	HTTP, FTP, TLS, SSL, HTTPS	HTTP/1.1	HTTP, CSS, JS
Caching	da	da	ne
Filtracija	da	da	ne
Kompresija	da	ne	da
Anonimnost	donekle	ne	ne

Tablica 2. Pregled besplatnih implementacija

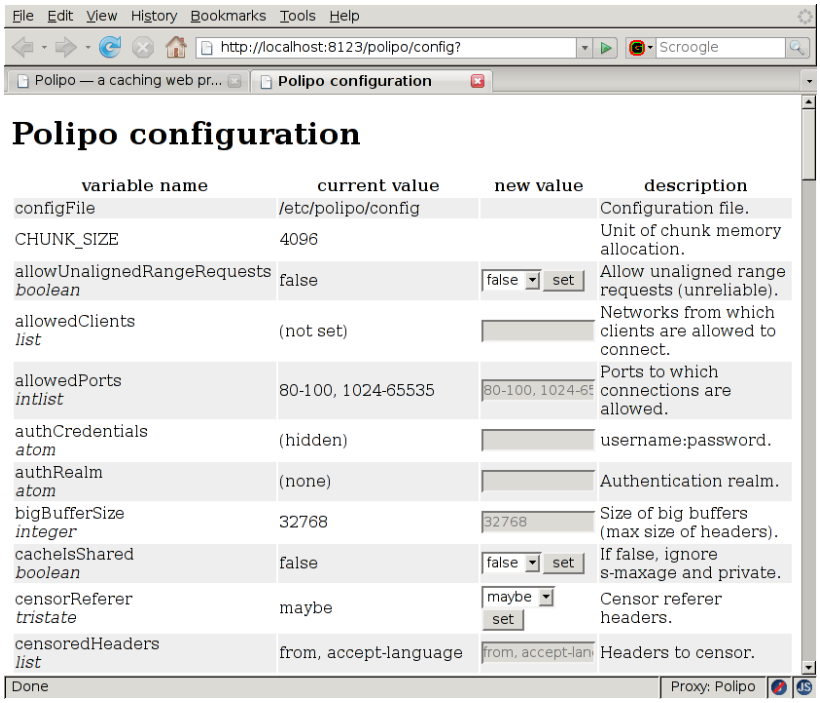
5.1. Polipo

Polipo je *proxy* poslužitelj namijenjen obradi zahtjeva nekolicine korisnika kojima omogućuje veće brzine dohvata spremanjem često dohvaćanog sadržaja u svoju lokalnu memoriju (eng. *caching*). Također, omogućuje jednostavnu filtraciju zahtjeva i podržava HTTP/1.1 pri čemu sve HTTP/1.0 zahtjeve klijenata prevodi u HTTP/1.1 i obratno, nakon dobivenog odgovora poslužitelja. Polipo je malen i jednostavan *proxy* poslužitelj koji ne zauzima puno resursa računala, izvodi se u pozadini, a može znatno ubrzati dohvat podataka s udaljenih poslužitelja.

Moguće ga je koristiti na sljedećim operacijskim sustavima:

- FreeBSD,
- NetBSD,
- OpenBSD,
- Mac OS X (verzije 10.2, 10.3.1 i 10.4),
- Microsoft Windows (sa instaliranim Cygwin programom).

Na slici 9 je prikazano korisničko sučelje Polipa, koje je jednostavno, ali omogućuje podešavanje raznih parametara poput parametara filtracije zahtjeva ili *caching*-a.



Slika 9. Polipo korisničko sučelje
Izvor: Google

5.2. Ziproxy

Ziproxy je još jednostavniji program koji ne izvodi *caching*, nego služi za kompresiju slike ili teksta te optimizaciju HTML/JS/CSS podataka. Može se koristiti i za uklanjanje reklama s web stranica te bilježenje prometa. Kao i Polipo, izvodi se u pozadini i ne troši puno računalnih resursa. Dostupan je na operacijskim sustavima Debian Linux, Gentoo Linux, Ubuntu Linux i FreeBSD za koje postoje detaljne upute za korištenje, ali ga je moguće koristiti na svim ostalim Unix/Linux operacijskim sustavima.

5.3. Squid

Squid je besplatni program otvorenog koda koji obavlja funkcije *proxy* poslužitelja. Ima razne primjene poput *caching*-a, kako bi se ubrzao promet, i filtracije prometa,. Može ga se postaviti iza klijentskih računala ili ispred poslužitelja, tj. može služiti kao *web* ili reverzni *proxy* poslužitelj. Primarno je namijenjen za operacijske sustave Unix/Linux , ali moguće ga je koristiti i na operacijskim sustavima Windows. S vremenom je Squid postao dostupan na velikom broju operacijskih sustava kao što se može vidjeti u tablici 3.

Skupina operacijskih sustava	Podržani operacijski sustavi
BSD	BDSI, DragonflyBSD, FreeBSD, GNU/kFreeBSD, Mac OS/X, NETBSD, NeXTStep, OpenBSD, SunOS
Linux	CentOS, Debian, Fedora, Gentoo, RedHat Enterprise Linux, Ubuntu
Unix	OSF/Digital Unix/Tru64, IRIX, SCO Unix, AIX, HP-UX
Windows	Windows 2000 Server, Windows NT, Windows XP Server, Windows 2003 Server, Windows Vista Server
Ostali	OS/2, Solaris

Tablica 3. Operacijski sustavi koje Squid podržava

Squid podržava veći broj protokola kao što su HTTP, FTP, TLS, SSL, Internet Gopher i HTTPS. Rad na njemu započeo je Duane Wessels. Dalje se razvijao na Kalifornijskom sveučilištu, a danas na njemu rade isključivo volonteri. Squid koriste administratori malih lokalnih mreža, ali i ISP-ovi (eng. *Internet Service Provider*) kako bi smanjili vrijeme potrebno za dohvat sadržaja s udaljenih poslužitelja i time poboljšali kvalitetu mreže.

Ako se Squid koristi kao *web proxy* poslužitelj postavlja se nakon korisnikovog računala tako da zahtjevi s korisnikova računala uvijek prolaze kroz proxy poslužitelj kako je prikazano na slici 2. On bilježi zahtjeve kako bi ubrzao dohvat često traženog sadržaja (eng. *caching*). Korisnikov preglednik dohvaća te podatke s *proxy* poslužitelja te se time smanjuje količina sadržaja koji je potrebno dohvatiti s udaljenog poslužitelja. Time se štedi vrijeme i novac. Squid donekle pruža zaštitu korisnika koji se spajaju preko njega jer udaljeni poslužitelji njega smatraju klijentom. S druge strane, Squid bilježi vrijeme i sadržaj zahtjeva koji kroz njega prolaze, preglednik i operacijski sustav korisnika, te URL stranice odakle zahtjev dolazi ako je to moguće vidjeti u HTTP_REFERER naredbi. Zbog toga je napadom na Squid poslužitelj moguće dobiti puno informacija o korisnicima koje se mogu iskoristiti za daljnje napade. Ipak, ova mogućnost je korisna ako se želi nadgledati promet koji korisnici izmjenjuju, kao što je to slučaj kod velikog broja tvrtki koje žele znati na što se njihov promet troši. Squid pruža nekoliko mogućnosti koje pomažu očuvanju korisnikove anonimnosti, najčešće skrivajući klijentsku IP adresu.

Ako se Squid koristi kao reverzni *proxy* poslužitelj, postavlja se prije poslužitelja. *Proxy* poslužitelj stoji između beskonačnog broja klijenata i konačnog broja poslužitelja. Podaci koje klijenti često traže se spremaju na *proxy* poslužitelj kako se ne bi morali stalno dohvaćati s poslužitelja. Time se poslužitelj rasterećuje. Poslužitelj sve zahtjeve dobiva s *proxy* poslužitelja i zato ne može točno pratiti promet što je ponekad potrebno zbog statistike. Da bi poslužitelj ipak došao do tih podataka potrebna je dodatna konfiguracija Squid *proxy* poslužitelja poput upotrebe zaglavlja HTTP_X_FORWARDED_FOR u kojem se zapisuje IP adresa klijenta koji je poslao zahtjev.

Ostale informacije o Squid poslužitelju se mogu naći na *web* stranici:

http://www.squid-cache.org/

6. Sigurnost *proxy* poslužitelja

Gotovo cijeli promet na Internetu svodi se na komunikaciju između klijenata i poslužitelja. Klijent šalje zahtjeve poslužitelju, a on klijentu šalje odgovore. Za uspješnu komunikaciju, klijent često šalje dodatne podatke o sebi. To može biti ime i inačica operacijskog sustava, ime i inačica Internet preglednika kao i neke njegove postavke. Ti podaci su potrebni kako bi poslužitelj mogao svoj odgovor uobličiti u *web* stranicu koju korisnikov Internet preglednik može prikazati na odgovarajući način. Ipak, u većini slučajeva ti podaci nisu potrebni jer se *web* stranica može prikazati samo u jednom obliku, pa su sve ove informacije redundantne.

Podaci koje klijent najčešće šalje poslužitelju su:

- ime i inačica operacijskog sustava,
- ime i inačica Internet preglednika,
- rezolucija u kojoj preglednik prikazuje *web* stranicu,
- podrška za Java/JavaScript,
- IP adresa klijenta i dr.

Najvažnija od ovih informacija je klijentska IP adresa iz koje se mogu saznati sljedeće informacije:

- država u kojoj se korisnik nalazi,
- grad u kojem se korisnik nalazi,
- korisnikov ISP i
- fizička adresa korisnika (za tvrtke).

6.1. Otkrivanje klijentske IP adrese iz HTTP zahtjeva

HTTP varijable iz kojih se može dobiti informacija o klijentskoj IP adresi su sljedeće:

- REMOTE_ADDR – korisnikova IP adresa.
- HTTP_VIA – adresa *proxy* poslužitelja ako se on koristi.
- HTTP_X_FORWARDED_FOR – korisnikova IP adresa ako se koristi *proxy* poslužitelj.

Sadržaj zadnje dvije varijable dodaje *proxy* poslužitelj kada zaprimi zahtjev. Ako se *proxy* poslužitelj ne koristi, sadržaj ovih varijabli bio bi sljedeći:

```
REMOTE_ADDR = IP_klijent
HTTP_VIA = neodređeno
HTTP_X_FORWARDED_FOR = neodređeno
```

Transparentni *proxy* poslužitelji nisu namijenjeni za skrivanje identiteta klijenata. Najčešće se koriste za preusmjeravanje klijentskih zahtjeva kako bi se zaobišla neka zabrana. Klijentska IP adresa se jasno može vidjeti, ali korisniku to neće smetati ako mu je jedina namjera da zaobiđe zabranu, a pritom mu je svejedno može li netko saznati da je on taj koji je zabranu zaobišao. *Proxy* poslužitelj mijenja varijable tako da se jasno vidi odakle je zahtjev došao (IP adresa klijenta) kao i da je prošao kroz *proxy* poslužitelj.

```
REMOTE_ADDR = IP_proxy
HTTP_VIA = IP_proxy
HTTP_X_FORWARDED_FOR = IP_klijent
```

S druge strane, anonimni *proxy* poslužitelji koriste se za zaobilaženje zabrana i skrivanje identiteta korisnika koji se njime služi. Anonimni *proxy* poslužitelji su zanimljivi ljudima koji ne žele postati metom marketinških akcija nakon što posjete neku *web* stranicu, ali i ljudima koji žele sakriti svoj identitet jer planiraju izvesti nešto nezakonito. Ljudima koji žele zadržati svoje pravo na slobodno razmišljanje, ali im država to ne dopušta blokirajući stranice koje smatra nepodobnima, ovakvi *proxy* poslužitelji su ne samo zanimljivi već i potrebni kako bi sakrili svoj identitet. Anonimni *proxy* poslužitelji dolaze u nekoliko oblika i nisu svi jednako pouzdani. Oni najjednostavniji zamjenjuju korisnikovu IP adresu s vlastitom pa varijable dobivaju sljedeći oblik:

```
REMOTE_ADDR = IP_proxy
HTTP_VIA = IP_proxy
HTTP_X_FORWARDED_FOR = IP_proxy
```

Ovakvi anonimni *proxy* poslužitelji su najčešći. Drugi način skrivanja korisnikove IP adrese je zamjena prave IP adrese s nasumičnom. Poslužitelj koji dobije zahtjev će misliti da zna odakle je zahtjev došao, a korisnik će ostati skriven. Sadržaj varijabli bi bio sljedeći:

```
REMOTE_ADDR = IP_proxy
HTTP_VIA = IP_proxy
HTTP_X_FORWARDED_FOR = nasumični_IP
```

Potrebno je primijetiti da u oba slučaja nije skrivena informacija da je korišten *proxy*. To može postavljati problem ako je odredišni poslužitelj postavljen tako da odbacuje sve zahtjeve koji dolaze s *proxy* poslužitelja. Takva zaštita se često postavlja kako bi se izbjegli napadi na poslužitelje koji dolaze upravo preko anonimnih *proxy* poslužitelja. Anonimni *proxy* poslužitelji koji skrivaju činjenicu da se koristi *proxy* poslužitelj nazivaju se elitnim *proxy* poslužiteljima. Varijable će imati sadržaj sličan slučaju kada se ne koristi *proxy* poslužitelj osim što će, umjesto korisnikove IP adrese, u varijabli REMOTE_ADDR stajati IP adresa *proxy* poslužitelja.

```
REMOTE_ADDR = IP_proxy
HTTP_VIA = neodređeno
HTTP_X_FORWARDED_FOR = neodređeno
```

6.2. Ostali načini otkrivanja klijentske IP adrese

Informacija o korisnikovoj IP adresi može se dobiti i na druge načine. Na prvi pogled, kolačići (eng. *cookies*) nisu povezani s *proxy* poslužiteljima. Oni se koriste za prenošenje malih količina informacija između klijenta i poslužitelja i nose dodatne informacije o klijentu. Ti podaci su pohranjeni u klijentskom Internet pregledniku, a *web* poslužitelj ih dohvaća kada su mu potrebne. Kolačići mogu biti automatski obrisani nakon što istekne sjednica u kojoj su korišteni ili mogu biti trajno pohranjeni na korisnikovom računalu. S kolačićima nije moguće otkriti korisnikovu IP adresu, ali poslužitelj može posumnjati na upotrebu *proxy* poslužitelja ako se IP adresa pohranjena u kolačiću i ona koja piše u HTTP zaglavlju ne podudaraju. Na taj način, *web* poslužitelj može otkriti čak i elitne *proxy* poslužitelje. Mnogi CGI *proxy* poslužitelji pružaju mogućnost da se prije njegove upotrebe isključi upotreba kolačića, te time spriječi *web* poslužitelj u otkrivanju elitnog *proxy* poslužitelja.

Ako *web* poslužitelj posumnja u upotrebu *proxy* poslužitelja jer upotreba kolačića nije bila isključena, klijentsku IP adresu može potražiti i pomoću skripta koje izvodi klijentski Internet preglednik. Skripte su jednostavni programi s ograničenom funkcionalnošću koji se izvode u pozadini, ali mogu saznati klijentsku IP adresu kao i ostale postavke Internet preglednika. Kao što je moguće spriječiti upotrebu kolačića, prije preusmjeravanja zahtjeva preko CGI *proxy* poslužitelja moguće je zabraniti i izvođenje JavaScript i VBScript skripti. Sličan problem može uzrokovati korištenje skripti pisanih u programskom jeziku Java, pa je puno jednostavnije u potpunosti isključiti izvođenje Jave u Internet pregledniku. ActiveX i razni dodaci Internet preglednika ne samo da mogu otkriti klijentsku IP adresu, već mogu i mijenjati postavke *proxy* poslužitelja. Zato je poželjno i njih isključiti prije upotrebe *proxy* poslužitelja.

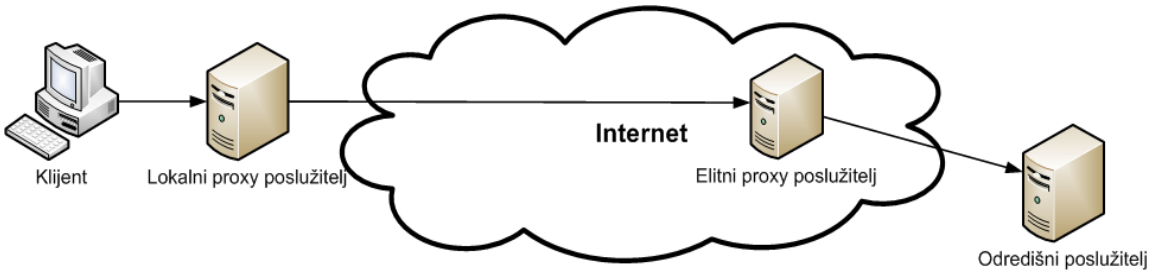
6.3. Rješenja za povećanje anonimnosti

Dakle, da bi klijentska IP adresa ostala sakrivena potrebno je isključiti sljedeće funkcionalnosti:

- Kolačići (eng. *cookies*),
- JavaScript/VBScript,
- Java i
- ActiveX kontrole.

Sve navedeno se može isključiti u postavkama Internet preglednika. Ako se koriste CGI *proxy* poslužitelji, kojima se pristupa preko *web* stranice, navedeno se može isključiti odabirom opcija na *web* stranici.

Problem je što se neke *web* stranice ne prikazuju ispravno ako su isključene skripte i kolačići. Ako se želi ostvariti anonimnost bez isključivanja skripti jedan način je prikazan na slici 10. Napravi se lokalna mreža u kojoj će sva računala imati lokalne IP adrese oblika 192.168.1.x ili slično tome (privatne IP adrese). Zahtjevi sa svih računala u mreži prolaze kroz *proxy* poslužitelj na izlazu iz lokalne mreže. On sve zahtjeve preusmjerava preko anonimnog (po mogućnosti elitnog) *proxy* poslužitelja. *Proxy* poslužitelj na izlazu lokalne mreže skriva sva računala u mreži i štiti ih od napada. IP adrese klijentskih računala koje se mogu pronaći u skriptama ili Java/ActiveX podacima su oblika 192.168.1.x i ne govore napadaču puno o klijentu. Ovaj način skrivanja klijentske IP adrese je dosta pouzdan, nije teško ostvariv, a korisnik i dalje može pregledavati stranice kao da se *proxy* poslužitelj ne koristi.



Slika 10. Dijagram sustava koji čuva anonimnost korisnika

Drugi način skrivanja klijentske IP adrese je korištenje SOCKS *proxy* poslužitelja i pripadnog softvera. On je puno sigurniji od ostalih *proxy* poslužitelja. Klijent se spaja na SOCKS *proxy* poslužitelj i prenosi mu adresu poslužitelja na koji se želi spojiti te podatke koje želi poslati. SOCKS *proxy* poslužitelj zatim stvara zahtjev s tim podacima i šalje ga do poslužitelja. Sa stajališta poslužitelja, SOCKS *proxy* poslužitelj je klijent i ne može nikako saznati IP adresu pravog klijenta osim ako ona nije zapisana u podacima koje je dobio. Često SOCKS *proxy* poslužitelj može stvoriti zahtjev tako da sakrije sve podatke o klijentu, pa poslužitelj niti iz tih podataka ne može saznati tko je pravi klijent.

Ipak, kao što je već spomenuto, upotreba elitnog *proxy* poslužitelja ne jamči apsolutnu anonimnost. Čak ni ulančavanje više anonimnih *proxy* poslužitelja neće sačuvati korisnikovu anonimnost, jer će informacije o korisniku uvijek ostati na prvom *proxy* poslužitelju do kojeg dolazi korisnikov zahtjev. Anonimnost se još više kompromitira ako se koriste *proxy* poslužitelji za čiju je upotrebu potrebna registracija.

7. Zaključak

Iz svega navedenog u ovom dokumentu može se zaključiti da su *proxy* poslužitelji vrlo korisni. Zbog svoje jednostavnosti vrlo su rašireni i koriste ih gotovo sve tvrtke, škole, javne ustanove, ali i veliki davatelji Internet usluga (ISP). Također, svaki korisnik može napraviti *proxy* poslužitelj na svom osobnom računalu koristeći neku od brojnih besplatnih implementacija *proxy* poslužitelja. Takvi *proxy* poslužitelji će ubrzati dohvaćanje sadržaja s udaljenih poslužitelja, povećati kvalitetu veze i smanjiti troškove jer se dohvaća manja količina podataka. Napredniji *proxy* poslužitelji mogu filtrirati zahtjeve klijenata i zabranjivati pristup nekim web stranicama ovisno o njihovom URL-u, sadržaju, protokolu koji koriste i sl. Ovakve zabrane su nekada opravdane, poput *proxy* poslužitelja koji brani pristup stranicama s pornografskim sadržajem korisnicima koji zahtjeve šalju s računala u knjižnici i sl. Ipak, u nekim državama se *proxy* poslužitelji koriste za provođenje cenzure blokirajući servise poput Internet tražilica i drugih poznatih *web* mjesta.

Zbog toga je nastala druga vrsta *proxy* poslužitelja koji imaju zadatak zaobilaženja ovih zabrana. To su *proxy* poslužitelji koji se koriste za tuneliranje. Metoda se temelji na činjenici da, iako je korisniku zabranjen pristup nekoj *web* stranici, *proxy* poslužitelju nije. Ako korisnik svoj zahtjev preusmjeri preko *proxy* poslužitelja, pod uvjetom da nema zabranu pristupa samom *proxy* poslužitelju, moći će pregledati zabranjenu *web* stranicu. Ova metoda se često koristi u državama koje provode Internet cenzuru, a zbog mogućih zakonskih posljedica zaobilaženja zabrana, korisnici žele očuvati svoju anonimnost. *Proxy* poslužitelji koji skrivaju podatke o korisniku zovu se anonimni *proxy* poslužitelji, a anonimnost čuvaju na različite načine. Neki načini su uspješniji od drugih, pa korisnik mora dobro proučiti *proxy* poslužitelj koji namjerava koristiti prije nego to učini. Anonimni *proxy* poslužitelji zanimljivi su i ljudima koji žele izvesti napad na neku *web* stranicu, a pritom žele ostati skriveni. Zato sve češće *web* poslužitelji provjeravaju dolaze li zahtjevi preko *proxy* poslužitelja i odbijaju pružanje uslugu istima. To predstavlja problem korisnicima koji ne planiraju izvesti napad nego samo žele pregledati stranicu koja im je inače zabranjena. Međutim, i ovaj način obrane *web* poslužitelja se može zaobići tako da će uvijek postojati borba između administratora *web* poslužitelja s jedne strane i napadača s druge, u kojoj su *proxy* poslužitelji samo jedan od alata za borbu.

8. Reference

[1] Wikipedia: Proxy server, http://en.wikipedia.org/wiki/Proxy_server

[2] Detailed proxy servers FAQ, http://www.freeproxy.ru/en/free_proxy/faq/index.htm

[3] Wikipedia: Squid, http://en.wikipedia.org/wiki/Squid_(software)

[4] Squid wiki, http://wiki.squid-cache.org/SquidFaq/AboutSquid

[5] A civisec project: Everyone's guide to by-passing Internet censorship, The University of Toronto, rujan 2007.