HRVATSKA AKADEMSKA I ISTRAŽIVAČKA MREŽA
CROATIAN ACADEMIC AND RESEARCH NETWORK

IPTables vatrozid

CCERT-PUBDOC-2007-07-198

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.

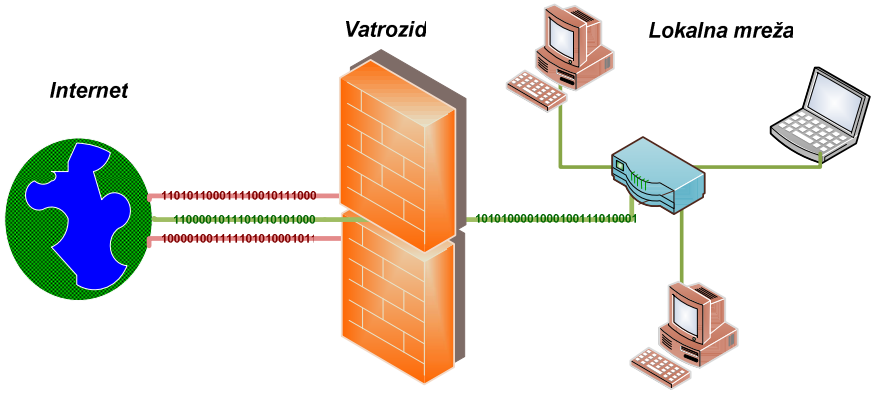
CARNet CERT, www.cert.hr - nacionalno središte za **sigurnost računalnih mreža** i sustava.

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

Za sigurnost svakog računala spojenog na Internet, važnost posjedovanja vatrozida nije upitna: ovaj programski proizvod je neizbježan s obzirom na učestalosti i razmjere napada zlonamjernih korisnika. Vrlo jednostavni vatrozidi implementirani su tako da na niskoj razini uspoređuju osnovne značajke paketa i na temelju određenih pravila poduzimaju odgovarajuće aktivnosti. Temeljne karakteristike su izvorna i ciljna adresa te priključci (tzv. mrežni portovi), a najvažnije aktivnosti su propuštanje i odbacivanje paketa. Napretkom Internet tehnologija, ovakav vatrozid postao je nedovoljan i neotporan na novije oblike napada. Stoga se uvode napredniji oblici vatrozidne zaštite koji pakete analiziraju na višoj razini i time pružaju precizniju i kvalitetniju zaštitu. Međutim, to ne znači da se prva grupa vatrozida - filtri mrežnih paketa - ne koriste i danas. Naprotiv, oni su toliko značajni da se ugrađuju u jezgre operacijskih sustava zajedno s drugim sigurnosnim i funkcionalnim mehanizmima i neizbježan su alat u borbi protiv internetskih prijetnji.
Vatrozid se može pojednostavljeno prikazati kao mehanizam upogonjen između lokalne mreže i Interneta, a dan je na slici ispod.



Slika 1: Shematski prikaz zaštite lokalne mreže pomoću vatrozida

Ovakva rješenja namijenjena su spajanju više računala na Internet i zadovoljavaju potrebe na profesionalnoj razini. U slučaju korisnika osobnih računala, ovakav oblik zaštite daleko je od praktičnog i prihvatljivog budući da bi zahtijevao jedno namjensko računalo koje bi imalo ulogu posrednika za pristup Internetu. To ujedno zahtijeva održavanje od strane stručnog osoblja ili edukaciju korisnika samo za ispravno postavljanje veze prema Internetu. Ovo rješenje nije prihvatljivo pa su razvijena druga rješenja u obliku programske potpore, jednostavna za korištenje i nadogradnju te kao takva idealna za korisnike osobnih računala. Prethodno prikazana shema sklopovskog vatrozida gotovo je istovjetna programskoj shemi pa se u detaljnija razmatranja neće ulaziti.
IPTables jedna je od poznatijih implementacija besplatnih vatrozida otvorenog koda, a namijenjena je Linux operacijskim sustavima. Vatrozid je prvenstveno orijentiran ka drugom sloju referentnog TCP/IP (eng. *Transmission Control Protocol / Internet Protocol*) modela, ali može koristiti i elemente trećeg sloja kao i većina današnjih IP filtara. Držeći se strogo definicije IP filtara i usmjerenosti na drugi sloj TCP stoga, mehanizam bi mogao samo filtrirati na temelju IP zaglavlja (izvorne i ciljne IP adrese, nekoliko parametara poput TOS (eng. *Type Of Service*) i TTL (eng. *Time To Live*)). No, kako je već napomenuto, ovi mehanizmi mogu koristiti i podatke drugih slojeva, a time i čitati TCP i UDP, pa čak i MAC zaglavlja.
Mehanizam praćenja uspostavljenih veza (eng. *connection tracking system*) omogućuje naprednije tehnike filtriranja, ali se ne radi o jednostavnom zapisivanju primljenih paketa. Takvo praćenje paketa, kao što je realizirano izvedbom programske podrške za komunikaciju TCP protokolom, koristilo bi previše resursa što nije prihvatljivo rješenje.
Dokument opisuje NetFilter projekt i u okviru njega projekt IPTables. Slijede opisi koncepta tablica i lanaca, neizbježni za shvaćanje čitavog mehanizma filtriranja. Zatim su opisane najčešće naredbe za stvaranje pravila uz odgovarajuće primjere, a dan je pogled i na pomoćne naredbe vezane uz

spremanje i obnovu pravila te rad sa samim servisom. Konačno, dan je konkretan i vrlo jednostavan primjer implementacije temeljnih pravila uz obrazloženje.

2. IPTables kao dio NetFilter projekta

NetFilter je projekt usmjeren na razvoj programske podrške vezane uz filtriranje mrežnih paketa, integrirane unutar jezgre operacijskog sustava, inačica 2.4.x i 2.6.x. Jedan od uradaka je i IPTables vatrozid. Organizacija posjeduje web sjedište na adresi http://netfilter.org/ gdje je moguće doći do detaljnijih podataka.

2.1. NetFilter

Programska podrška NetFilter, pored mehanizama za filtriranje paketa, implementira i mehanizme za NAT adresiranje te izmjenu mrežnih paketa. Nasljednik je sustava sa starijih inačica jezgri:

- *ipchains* (Linux 2.2.x) i
- *ipfwadm* (Linux 2.0.x).

Netfilter je skupina programskih funkcija jezgre operacijskog sustava koja omogućuje jezgrinim modulima registriranje tzv. *callback* funkcija u okviru mrežnog stoga (eng. *network stack*). Te funkcije izvode se, odnosno pozivaju sinkrono s nekim događajem s kojim su registrirane. U ovom slučaju, funkcije se pozivaju pri pojavi svakog novog mrežnog paketa.
IPTables mehanizam koristi tablične strukture za zapisivanje pravila. Svako pravilo u okviru tablice sastoji se od stanovitog broja elemenata za usporedbu i akcije koja se odvija u slučaju uspješne usporedbe.
Podsustavi *netfilter*, *ip_tables*, *ip_conntrack*, *nf_conntrack* i *NAT* čine glavne komponente cjelokupnog projekta.
Odlike ovog programskog paketa su:

- statičko, tzv. *stateless* filtriranje mrežnih paketa (IPv4 i IPv6) ;
- naprednije dinamičko, tzv. *stateful* filtriranje mrežnih paketa (IPv4 i IPv6) kod koga se prate uspostavljene veze;
- NAT/NAPT (eng. *Network Address [Port] Translation*);
- proširiva programska infrastruktura;
- višeslojan API (eng. *Application Programming Interface*) za korištenje modula drugih proizvođača i
- velik broj programskih priključaka i modula.

Temeljem netfilter/iptables paketa moguće je:

- izgraditi specifične vatrozid aplikacije prema korisničkim zahtjevima temeljene na *stateless* i *stateful* filtriranju;
- moguće je koristiti NAT adresiranje za razvoj alata namijenjenih dijeljenju Internet veza, odnosno spajanju mreža na Internet korištenjem jedne javne adrese;
- implementirati posredničke poslužitelje (eng. *proxy*);
- razviti pomoćni mehanizam za izgradnju složenijih sustava za usmjeravanje i kvalitetu usluge (eng. QoS - *Quality of Service*) te
- vršiti izmjene nad paketima koje obuhvaćaju promjene TOS (eng. *Type Of Service*) i TTL (eng. *Time To Live*) te drugih polja iz zaglavlja paketa.

2.2. IPTables

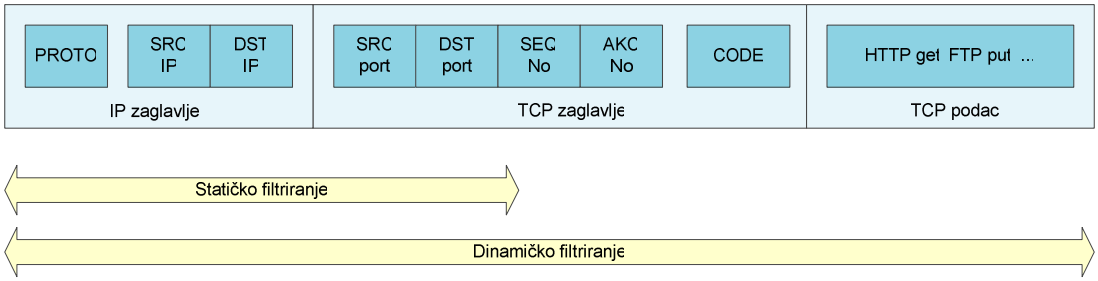
IPtables je program koji se koristi za postavljanje pravila vezanih uz filtriranje paketa, a dostupan je za jezgre operacijskih sustava inačica 2.4.x i 2.6.x. Alat je namijenjen prvenstveno administratorima sustava, a kao sučelje se koristi naredbeni redak. Postoje brojna rješenja koja omogućavaju jednostavniju izgradnju pravila, ali većina tih korisnicima jednostavnijih alata ne omogućava napredne mogućnosti konfiguracije vatrozida kao što to IPTables omogućava.
Budući da je prevođenje adresa (NAT) usko vezano uz podešavanje pravila filtriranja, razumljivo je i uključivanje NAT mehanizma u paket IPTables.
Paket također uključuje ip6tables, aplikaciju namijenjenu filtriranju šeste inačice IP protokola - IPv6.
Temeljne odlike aplikacije obuhvaćaju:

- ispis pravila,
- dodavanje, brisanje i uređivanje pravila te
- ispis i poništavanje brojača iskorištenosti pojedinog pravila.

Tekuća inačica izvornog koda dostupna je korištenjem repozitorija na adresi https://svn.netfilter.org/netfilter/trunk/iptables/.
Autori IPTables paketa su stručnjaci iz razvojnog tima NetFilter projekta, ali tijekom posljednjih godina mnogo je članova širom svijeta pomoglo u napretku tog paketa.
Temeljni razvojni tim čine: Marc Boucher, Martin Josefsson, Jozsef Kadlecsik, Patrick McHardy, James Morris, Harald Welte i Rusty Russell.

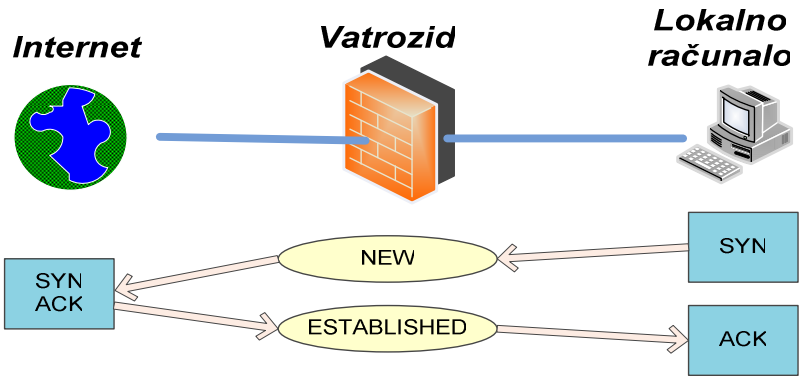
2.3. Dinamičko filtriranje

Vrlo važna funkcionalnost IPTables paketa je dinamičko filtriranje (eng. *stateful inspection*). Statičko filtriranje temelji se na provođenju usporedbe na pojedinačnom paketu, bez obzira koja je njegova uloga u komunikaciji. Kao što je već spomenuto u uvodnom dijelu, kod statičkog mehanizma radi se o ispitivanju izvorne i ciljne adrese i priključka te određenih polja paketa. Ovo je efikasan način filtriranja, ali ograničenih mogućnosti. Zbog važnih karakteristika paketa, kao što su pripadnost određenoj vezi (konekciji), pokazalo se potrebnim uvođenje novog mehanizma kojim bi se moglo odrediti i neke druge elemente potrebne za donošenje odluke o prihvaćanju ili odbacivanju paketa.
Korištenjem odgovarajućih tablica, zapisuju se podaci o uspostavljenim vezama i to od početka uspostave veze sve do njezinog završetka. Tijekom filtriranja, pored tablica s pravilima statičkog filtriranja, koriste se i podaci iz opisanih tablica te se na taj način izvodi dinamičko filtriranje.
Razliku između značajnih dijelova mrežnog paketa za pojedini način rada, pregledno je moguće analizirati sljedećom shemom.



Slika 2: Odnos statičkog i dinamičkog filtriranja

Sam mehanizam koristi najviše resursa kod analize paketa namijenjenih uspostavljanju veze, a daljnje određivanje pripadnosti paketa nekoj vezi računski je manje zahtjevno.
Tijek rada dinamičkog uspoređivanja paketa zasniva se na uspostavljanju veze u tri koraka (eng. *three-way handshake*). Klijent, kod pokušaja uspostavljanja veze, najprije šalje paket s postavljenom SYN zastavicom u zaglavlju paketa. Svi paketi s ovom zastavicom kod IPTables mehanizma dobivaju atribut NEW. Ako je druga strana dostupna, ona odgovara povratnim paketom koji ima postavljene zastavice SYN i ACK.



Slika 3: Stanja kod uspostave veze

Konačno, slijedi odgovor klijenta slanjem paketa s postavljenom ACK zastavicom. Sada je omogućeno funkcioniranje vatrozida kod kojeg se propuštaju svi izlazni paketi, a prihvaćaju samo ulazni koji su dio uspostavljene veze. Na taj način sprečava se otimanje sjednice ubacivanjem posebno oblikovanih mrežnih paketa. Za završetak veze odvija se sličan niz postupaka i uvodi se stanje CLOSED. UDP pakete prati se na sličan način, uz nešto modificiranu proceduru završetka veze.

Određene protokole teže je pratiti, a tu se između ostalih ubrajaju ICQ, IRC i FTP te TFTP. Kao primjer može poslužiti uspostava veze kod FTP protokola. Kod aktivne uspostave poziva, klijent šalje poslužitelju novi broj priključka na kojemu će se odvijati daljnja komunikacija. Priključak se odabire nasumično iz intervala od 1025 do 65535, a poslužitelj uspostavlja vezu korištenjem odabranog priključka. Upravo je ovdje problem za mehanizam praćenja veze koji ni na koji način ne može znati koji će se priključak koristiti. Zbog toga se upotrebljavaju pomoćni mehanizmi koji pretražuju pakete kako bi odredili potrebne informacije. Kada se pronađu takve informacije, paketi tih veza se označavaju posebnom oznakom RELATED kojom se omogućuje nesmetana daljnja komunikacija.Postoji još i oznaka INVALID kojom se označavaju svi paketi koje nije moguće drugačije klasificirati temeljem njihovog sadržaja, odnosno sadržaja u odgovarajućim tablicama.

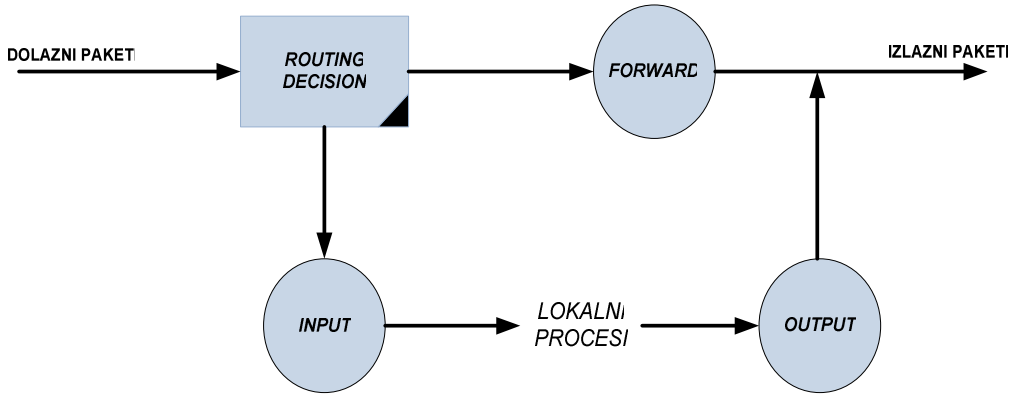
3. Koncepti tablica i lanaca

U nastavku poglavlja raspoloživ je opis putova - lanaca, koje slijede mrežni paketi prolaskom kroz mehanizam IPTables vatrozida. Svaki segment tog puta obilježen je odgovarajućom tablicom čiji je način korištenja posebno važan kod stvaranja pravila.

3.1. Lanci i akcije

Izraz lanac (eng. *chain*) odnosi se na označenu točku kroz koju u određenom trenutku prođe svaki paket i gdje se uspoređuje s nizom pravila koja određuju daljnji tok tog paketa. Radi se o pravilima kojima se određuje logički slijed aktivnosti - lanac, čije postavljanje se obavlja naredbom `iptables (8)`.

Pojednostavljeni prikaz čitavog IPTables mehanizma dan je na sljedećoj slici.



Slika 4: Pojednostavljen prikaz lanaca u iptables mehanizmu

Tri kruga prikazana na prethodnoj slici označavaju redom lance:

- *FORWARD*,
- *INPUT* i
- *OUTPUT*.

Za svaki paket koji pristigne na neko od mrežnih sučelja, serijsko ili neko drugo, najprije se određuje njegovo odredište (eng. *Routing decision*). Ukoliko je krajnji cilj paketa neki lokalni proces na istom računalu na kojemu je upogonjen vatrozid, paket se usmjerava prema *INPUT* lancu. Ako se pokaže da je odredište paketa neko drugo računalo, tada ga se prosljeđuje *FORWARD* lancu. Konačno, paket kojeg generiraju neke lokalne aplikacije odnosno procesi na lokalnom računalu, usmjerava se *OUTPUT* lancu u kojem se odlučuje što je potrebno napraviti s tim mrežnim paketom generiranim od strane samog vatrozida.

Nekoliko je akcija primjenjivo nad svakim paketom u navedenim točkama od kojih su najznačajnije sljedeće:

- *DROP*,
- *ACCEPT*,
- *QUEUE*
- *RETURN*,
- *REJECT* i
- *LOG*.

Akcija *DROP* označava odbacivanje paketa na način da ga se jednostavno zanemari pri čemu se ne šalje nikakva obavijest o odbačenom paketu. *ACCEPT* podrazumijeva prosljeđivanje paketa u sljedeći segment koji dolazi prema jednostavnoj shemi prikazanoj na slici *Slika 4*. *QUEUE* označava prosljeđivanje paketa korisničkom procesu iz memorijskog prostora NetFilter mehanizma u memorijski prostor korisničkih aplikacija. *RETURN* obilježava prestanak usporedbe u tekućem lancu i povratak na sljedeću neobrađenu usporedbu u prethodnom lancu. *REJECT* slično kao i *DROP* odbacuje paket, ali na način da pošiljatelja obavijesti o tome. *LOG* se koristi za zapisivanje potankosti o paketima preuzetih iz zaglavlja paketa i drugih potencijalno važnih informacija.

Prema dosad rečenom, očito je kako lanac predstavlja niz pravila. Svako pravilo ima oblik

```
AKO zaglavlje_paketa ODGOVARA pravilu ONDA akcija
```

i zapisano je u odgovarajućoj tablici.

Počevši od prvog pravila u nizu, ako tekući paket ne zadovoljava dani uvjet, nastavlja se s provjerom svakog sljedećeg pravila. U slučaju da paket zadovoljava neko pravilo, upravo se ono primjenjuje za određivanje sudbine paketa i zavisno o definiranoj akciji ovisi da li će se daljnji tijek ispitivanja preostalih pravila prekinuti. Npr. ukoliko je akcija logiranja mrežnog paketa tada se daljnji tijek ispitivanja preostalih pravila neće nužno prekinuti.

U protivnom slučaju, kada se ne nađe pravilo koje odgovara stanju iz paketa, primjenjuje se opće pravilo lanca tzv. *chain policy* u određivanju sudbine paketa. Opće pravilo lanca predstavlja vrlo važan element koji određuje razinu opće sigurnosti. Opće pravilo može uključivati ili prihvaćanje ili odbacivanje svih paketa za koje se ne pronađe odgovarajuće pravilo. U slučaju prihvaćanja paketa,

prednost je ta da dizajner pravila ne mora brinuti o eventualnom zaboravljanju navođenja nekog od pravila. Ukoliko se pojavi paket koji ne zadovoljava pravila, prema općem pravilu on biva prihvaćen i nema rizika od odbacivanja nekog važnog nepredviđenog paketa. Ovo je princip tzv. crne liste (eng. *black list*) prema kojemu se na kraju svih usporedbi mrežni paket prihvaća, ali su prije tog „općeg" prihvaćanja uobičajeno navedene različite naredbe neprihvaćanja određenih mrežnih paketa. Te naredbe koje ne prihvaćaju određene mrežne pakete predstavljaju tzv. crnu listu nepoželjnih mrežnih paketa.
U suprotnom, ukoliko se odabere opće pravilo odbacivanja paketa, povećava se sigurnost sustava u velikoj mjeri budući da nepredviđeni paketi nikada neće doći u sustav. Ovaj način rada predstavlja princip tzv. bijele liste (eng. *white list*) prema kojemu se na kraju svih usporedbi mrežni paket odbacuje, ali su prije tog „općeg" odbacivanja uobičajeno navedene različite naredbe prihvaćanja određenih prihvatljivih mrežnih paketa.
Očito je prednost jednog načina ujedno i mana drugog, ali za povećanu sigurnost računalne mreže savjetuje se korištenje pravila podrazumijevanog odbijanja paketa.

Kao što je već spomenuto u ovom poglavlju, po ulasku paketa u sustav preko nekog mrežnog sučelja najprije slijedi određivanje puta temeljeno na njegovu konačnom odredištu. Ako je konačno odredište lokalno računalo, paket se usmjerava prema *INPUT* lancu. Ako ga, temeljem pravila, lanac propusti, lokalni proces primit će taj paket. U drugom slučaju, kada paket nije namijenjen procesu na lokalnom računalu, a ako uz to i jezgra operacijskog sustava nema omogućeno prosljeđivanje IP paketa ili ne zna na koji način proslijediti paket, isti biva odbačen. Ako je mehanizam prosljeđivanja ispravan, paket ide ka *FORWARD* lancu i ako zadovolji pravila dana tim lancem, biva proslijeđen dalje korištenjem odgovarajućeg mrežnog sučelja. Konačno, i proces pokrenut na lokalnom računalu može biti izvorište mrežnih paketa. U tom slučaju koristi se *OUTPUT* lanac, a paket biva odaslan na mrežu tek ukoliko zadovolji pravila iz tog lanca.
Za ilustraciju, slijedi primjer rada sljedeće jednostavne naredbe:

```
[user@localhost]# ping 127.0.0.1
```

Program `ping` namijenjen je slanju tzv. *echo* paketa odredištu danom IP adresom. Adresa 127.0.0.1 naziva se i *loopback* odnosno povratnom adresom jer svi paket poslani s nekog računala na tu adresu dolaze na to isto izvorišno računalo. Ideja naredbe je slanje paketa vlastitom računalu i tijekom toga odvija se neprekidan ispis uspješnosti slanja zahtjeva i primanja odgovora.
Prije izmjene bilo kakvih pravila poželjno je pregledati trenutno stanje lanaca sljedećom naredbom:

```
# iptables -L
```

Kao rezultat naredbe dobiva se ispis:

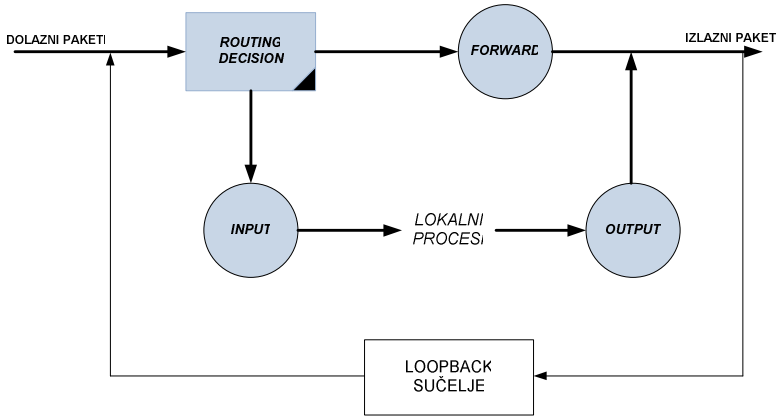
```
Chain INPUT (policy ACCEPT)
target prot opt source destination

Chain FORWARD (policy ACCEPT)
target prot opt source destination

Chain OUTPUT (policy ACCEPT)
target prot opt source destination
```

U drugom naredbenom retku treba pokrenuti sljedeću naredbu:

```
# iptables -P INPUT DROP
```

Po izvođenju naredbe može se uočiti da je naredba `ping` prestala ispisivati odgovore. Razlog tome je sljedeći. Naredba `ping` šalje pakete koji dolaze u *OUTPUT* lanac, a podrazumijevana akcija je prihvaćanje paketa odnosno njihovo prosljeđivanje. Sučelje 127.0.0.1 razvijeno je tako da svaki odlazni paket proslijedi ulaznom sučelju, čime se simulira dolazak paketa s nekog drugog računala.

Nakon usmjeravanja, isti se paket pojavljuje u ulaznom lancu *INPUT*. U početku njegova vrijednost je *ACCEPT* i zato paketi uspješno prolaze. Nakon obavljanja prethodno navedene naredbe, dolazi do promjene podrazumijevane vrijednosti lanca u odbacivanje paketa i prema tome svaki paket uključujući i one generirane `ping` naredbom biva odbačen. U tom slučaju nema nikakvog ispisa. Sljedećim prikazom ilustrirana je upravo opisana situacija.



Slika 5: Lanci i povratno (loopback) sučelje

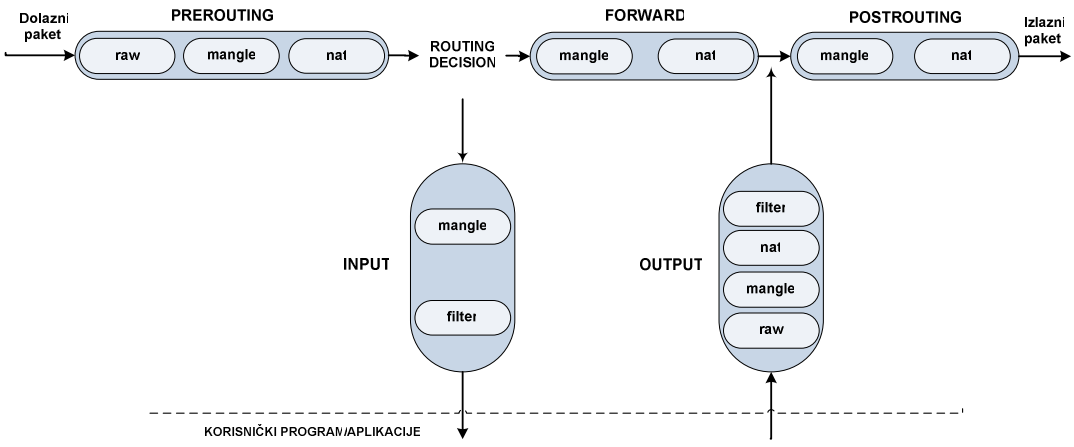
Povratak na prethodno stanje moguće je izvesti naredbom:

```
# iptables -P INPUT ACCEPT
```

Za usmjerivače općenito vrijedi slično razmatranje s tim što se pravila trebaju odnositi na *FORWARD* umjesto *OUTPUT* lanca.

3.2. Tablice i ostali lanci

U prethodnom poglavlju dani su temelji vezani uz lance u mehanizmu IPTables vatrozida. Slijedi opis pripadnih tablica, ali i naprednijih lanaca koji nisu opisani u prethodnom odjeljku.



Slika 6: Detaljniji prikaz lanaca i tablica

Prema slici Slika *6* mogu se uočiti još dva lanaca pored već spomenutih *INPUT*, *OUTPUT* i *FORWARD*. Riječ je o lancima:

- *PREROUTING* i
- *POSTROUTING*

koji se koriste kod tehnike prevođenja adresa (eng. *NAT – Network Address Translation*). Prvi se najčešće koristi neposredno prije donošenja odluke o usmjeravanju paketa za prevođenje ciljnih adresa (eng. *DNAT – Destination Network Address Translation*). Drugi se, između ostalog, koristi

neposredno prije slanja paketa na izlazno sučelje za prevođenje izvornih adresa (eng. *SNAT – Source Network Address Translation*). Osim novih lanaca, moguće je primijetiti i postojanje elemenata unutar svakog lanca - po dva, tri i četiri elementa. Ovi elementi nazivaju se tablicama, a njihov redoslijed na shematskom prikazu važan je kod razmatranja njihovog funkcioniranja budući da se tablice ispituju onim redoslijedom kojega impliciraju strelice na shemi. Primjerice, kod *OUTPUT* paketa, najprije se provjeravaju pravila iz tablice *raw*, potom *mangle*, zatim iz tablice *nat* i konačno iz tablice *filter*. Odabir tablice iz određenog lanca u koju se želi zapisati pravilo može se učiniti se na sljedeći način.

```
# iptables -A PREROUTING -t nat ...
```

Ukoliko tablica nije navedena, podrazumijevana za sve lance osim PREROUTING i POSTROUTING je uvijek *filter*. Budući da u spomenutim lancima ne postoji tablica *filter*, za zapisivanje u tablice tih lanaca potrebno je koristiti parametar `-t` uz naziv željene tablice.

Postojeće tablice su sljedeće:

- *filter* - određivanje prihvaćanja ili odbijanja paketa,
- *mangle* - izmjena pojedinih polja paketa ukoliko je to potrebno,
- *nat* - prevođenje IP adresa i priključaka te
- *raw* - ispitivanje paketa prije nego ga obuhvati mehanizam praćenja veze.

3.2.1. Tablica *filter*

Najčešće se koristi za filtriranje paketa. U nju se zapisuju pravila koja paketi moraju zadovoljiti da bi bila poduzeta odgovarajuća akcija, također zapisana u istoj tablici. Ovdje se paketi analiziraju i na temelju sadržaja odbacuju ili prihvaćaju. Iako se u tablicu mogu zapisati sve akcije temeljene na karakteristikama paketa, pravilo je da se prvenstveno tablica popunjava akcijama vezanim uz filtriranje.

3.2.2. Tablica *nat*

Namjena ove tablice je podešavanje pravila vezanih uz prevođenje adresa (eng. *NAT - Network Address Translation*). Preciznije rečeno, tablicu treba koristiti samo za izmjene na paketima koje se tiču izvorne i ciljne IP adrese. Također, izmjene nije moguće raditi samo na IP adresama već je to isto moguće raditi i na mrežnim priključcima (eng. *ports*).
Akcije koje se mogu poduzeti korištenjem *nat* tablice su:

- *DNAT* (*Destination NAT*),
- *SNAT* (*Source NAT*),
- *MASQUERADE* i
- *REDIRECT*.

DNAT se koristi kod prevođenja ciljnih adresa, a najčešće je to prevođenje javne IP adrese u privatnu. Ovo je praktično u slučaju kada se, primjerice, web poslužitelj nalazi u okviru privatne računalne mreže, a time i privatnog IP segmenta nedostupnog s Interneta. Prepisivanjem ciljne adrese na računalu koje posjeduje sučelje s javnom IP adresom, omogućena je komunikacija vanjskog svijeta s navedenim poslužiteljem.
SNAT se koristi za izmjene izvornih adresa paketa. Sličan primjer kao i u prethodnom, može poslužiti za ilustraciju i u ovom slučaju. Neka bilo koje računalo u privatnoj mreži želi komunicirati s proizvoljnom točkom na Internetu. Računalo posjeduje privatnu IP adresu, za primjer se može uzeti adresa 192.168.1.5. Vrijedi pravilo da se privatne IP adrese ne smiju pojaviti na Internetu budući da IANA (eng. *Internet Assigned Numbers Authority*) standard određuje taj segment privatnim i za njega ne postoje mehanizmi usmjeravanja na Internetu. U slučaju paketa s izvornom IP adresom iz segmenta, primjerice 192.168.0.0/24, paket se nikada ne bi vratio na izvorište. Zbog toga je potrebno na točki koja spaja 'vanjski svijet' i privatnu mrežu omogućiti prevođenje izvorne privatne adrese u javnu IP adresu kako bi se povratna informacija mogla vratiti do računala s javnom adresom, a time i do izvornog računala koje je odaslalo upit.

DNAT i *SNAT* omogućuju i pretvorbu ciljnog odnosno izvornog priključka. Također, promjene IP adresa je moguće izvesti slučajnim odabirom IP adresa iz zadanog raspona, a isto vrijedi i za priključke. Ovaj se mehanizam može koristiti, primjerice, za raspoređivanje posla u situaciji kada se IPTables koristi za prosljeđivanje paketa na više poslužitelja koji izvode isti zadatak.
MASQUERADE akcija se koristi u iste svrhe kao i *SNAT*, ali izvođenje ove akcije nešto je zahtjevnije zbog provjeravanja korištene IP adrese pa se ne savjetuje korištenje umjesto *SNAT* akcije. Radi se o tome da se kao izvorna adresa paketa koristi IP adresa mrežnog sučelja, bez obzira koja ona bila, tj. moguće ju je tijekom rada i mijenjati. Na ovaj način moguće je koristiti DHCP (eng. *Dynamic Host Configuration Protocol*) protokol dinamičkog dodjeljivanja adresa jer *SNAT* mehanizmom to nije omogućeno.
REDIRECT akciju je moguće koristiti jedino u *PREROUTING* i *OUTPUT* lancima u okviru *nat* tablice, a namjena joj je usmjeravanje paketa na lokalno računalo. Riječ je o tome da se ciljna adresa jednostavno prepiše adresom lokalnog računala, a zadavanjem naredbe moguće je eksplicitno navesti i željeni ciljni priključak, niz priključaka od kojih se slučajnim odabirom uzima jedan ili ostaviti postojeći priključak zapisan u paketu. Primjena ove akcije je kod posredničkih poslužitelja (eng. *proxy*) kada se njihovo korištenje želi učiniti transparentnim.

3.2.3. Tablica *mangle*

Namijenjena je izvođenju aktivnosti koje mijenjaju sadržaj paketa. Moguće je promijeniti polja poput TOS (eng. *Type of Service*) i još nekih opisanih u nastavku. Korisnike se strogo opominje da ne koriste ovu tablicu ni za kakve oblike filtriranja niti prevođenja adresa kao što su *DNAT*, *SNAT* i *MASQUERADING*.
Sljedeće akcije mogu se koristiti samo u tablici *mangle* i ni u jednoj drugoj tablici IPTables vatrozida.

- *TOS*,
- *TTL*,
- *MARK*,
- *SECMARK* i
- *CONNSECMARK*.

TOS se koristi za izmjenu polja *Type Of Service* u okviru mrežnog paketa, a polje određuje načine prosljeđivanja paketa na mreži u ovisnosti o cijeni, brzini i pouzdanosti dostupnih kanala. Ovakva akcija se poduzima kod postavljanja mrežnih pravila vezanih uz usmjeravanje i dr. Međutim, velik broj usmjerivača ne temelji svoje odluke na ovom polju pa treba primijetiti da ovakvi paketi mogu biti pogrešno protumačeni.
TTL akcija koristi se za izmjenu parametra *Time To Live*. TTL parametar označava duljinu života paketa ili broj ponovnih odašiljanja nakon kojeg se paket zanemaruje. Ova akcija korisna je mogućnost kada se koriste davatelji Internet usluga (eng. *ISP - Internet Service Provider*) koji ne odobravaju spajanje više računala na Internet korištenjem jedne linije. Naime, različita TTL polja jedan su od najjednostavnijih pokazatelja da više računala koristi istu liniju, pa se ovom akcijom to može korigirati.
MARK se koristi za dodavanje posebnih oznaka paketima, a ta polja imaju značaj jedino kod naprednijih izvedbi usmjerivača gdje se i ono uzima u obzir prilikom usmjeravanja.
SECMARK omogućuje dodavanje oznaka vezanih uz sigurnost, ali isto ovisi o implementaciji usmjerivačkih programa odnosno uređaja.
CONNSECMARK se koristi za kopiranje sigurnosnih oznaka jednog paketa na sve pakete iz veze.

3.2.4. Tablica *raw*

Namjena tablice *raw* je označavanje paketa nad kojima se želi onemogućiti rad sustava za praćenje veze (eng. *connection tracking system*). Ovo se izvodi korištenjem NOTRACK akcije nad paketom. Tablica se nalazi isključivo u okviru *PREROUTING* i *OUTPUT* lanaca. U ostalim lancima implementacija raw tablice niti nema smisla jer ovo su jedine točke u koje paket pristiže prije nego ga obuhvati mehanizam praćenja veza. Budući da praćenje veza koristi stanovitu količinu resursa, poželjno ga je u najvećoj mogućoj mjeri umanjiti. Za primjer se može uzeti rad usmjerivača kroz kojeg prolaze velike količine mrežnog prometa. Umanjiti opterećenje moguće je uklanjanjem praćenja paketa koji nisu namijenjeni samom usmjerivaču. Za ove potrebe koristi se tablica *raw* kojoj se dodaje pravilo da se svi

paketi namijenjeni lokalnom računalu trebaju prihvatiti (ACCEPT), a svima ostalima treba postaviti oznaku NOTRACK. Na taj način vršiti će se praćenje onih paketa za koje to ima smisla činiti, a pored toga će se računalo osloboditi nepotrebnog opterećenja.

4. IPTables naredbe i akcije

Jednostavni primjeri korištenja IPTables mehanizma dani su u prethodnom poglavlju. Slijedi pojašnjenje uobičajenog podešavanja vatrozida, a zatim i kratak opis nešto naprednijih mogućnosti.

4.1. Naredbe i akcije

Parametar `-j` odnosno `--jump` određuje akciju koja će se izvesti nakon usporedbe s danim pravilom. U prethodnom poglavlju vezanom uz lance i tablice navedene su i opisane moguće akcije. Akcija može uključivati slanje paketa na lanac koga je korisnik stvorio, može biti jedna od ugrađenih akcija kojom se određuje sudbina paketa ili proširenje (eng. *extension*). Ukoliko se ne navede ovaj parametar, u slučaju kada paket zadovoljava elemente navedene u pravilu, ništa se s njim neće dogoditi u smislu određivanja njegove sudbine - uvećati će se samo brojač (eng. *counter*) vezan uz to pravilo.
Parametrom `-g` odnosno `--goto` izravno se određuje korisnički stvoren lanac kojemu se paket šalje.
Za razliku od usmjeravanja tijeka obrade korištenjem naredbe `--jump`, u ovom slučaju pojava *RETURN* specifikacije u odredišnom lancu neće vratiti izvođenje usporedbe u lanac s kojeg je preusmjeravanje izvršeno `--goto` parametrom nego na posljednji lanac s kojeg je izvođenje usmjereno `--jump` parametrom. Sljedeći ispis pokazuje nekoliko primjera zadavanja naredbi korištenjem opisanih opcija. Prva naredba predstavlja prihvaćanje odlaznog mrežnog prometa. Druga naredba predstavlja odbacivanje mrežnog prometa s izvorne IP adrese 192.168.1.101. A treća naredba predstavlja preusmjeravanje mrežnih paketa na korisnički kreirani lanac.

```
# iptables -A OUTPUT --jump ACCEPT
# iptables -A INPUT -s 192.168.1.101 -j REJECT
# iptables -A INPUT -p TCP -g NoviLanac
```

4.2. Određivanje izvorne i ciljne adrese

Izvorne i ciljne adrese mogu se prikazati na nekoliko načina. Dva načina obuhvaćaju određivanje računala simboličkim imenom (eng. *hostname*) ili IP adresom prema primjerima:

1. www.google.hr,
2. 209.85.135.103

Sljedeća dva odnose se na adresiranje mrežnog segmenta pri čemu je mrežnu masku (eng. *netmask*) moguće odrediti na dva načina - brojem nepromjenjivih bitova i IP adresom koja predstavlja nepromjenjive bitove:

3. 192.168.1.0/24
4. 192.168.1.0/255.255.255.0

Parametri kojima se određuje izvorna ili ciljna adresa su `-s` i `--source` te `-d` i `--destination`.
Posljednji način podrazumijeva upisivanje IP adresa u obliku danim sljedećim primjerom:

5. 192.168.1.5-192.168.1.10

U ovom slučaju koristi se posebna opcija `--src-range` za izvorne adrese i `--dst-range` za ciljne. Međutim, za korištenje tih opcija potrebno je uključiti odgovarajući modul korištenjem opcije `-m iprange`.
Određivanje svih raspoloživih IP adresa u okviru izvorne adrese paketa izvodi se oznakom 0/0 i može izgledati kao na sljedećem primjeru.

```
# iptables -A INPUT -s 0/0 -j DROP
```

Oznaka se rijetko koristi budući da se adresiranje svih IP adresa obavlja ispuštanjem izvorišta ili odredišta (u prethodnom primjeru izvorišta):

```
# iptables -A INPUT -j DROP
```

Naredba za određivanje svih paketa koji dolaze s IP adrese 209.85.135.103, a usmjereni su na neko računalo iz mrežnog segmenta 192.168.1.0/24 mogla bi se napisati u obliku

```
# iptables -s 209.85.135.103 -d 192.168.1.0/24 ...
```

Primjer naredbe koja koristi interval IP adresa za određivanje ciljnih adresa:

```
# iptables -A INPUT -p tcp -m iprange --dst-range 192.168.1.5-192.168.1.10
```

4.3. Određivanje invertnog pravila

Ponekad je neko pravilo jednostavnije i preglednije izreći korištenjem neželjenog pravila i njegovim invertiranjem. Korisnik IPTables mehanizma na raspolaganju ima oznaku `!` koja se čita NE (eng. *not*), a koristi se za negiranje željenog izraza.
Primjerice, neka je zadatak odrediti sve pakete koji nisu s lokalnog računala. Očito je kako bi bez korištenja negacije ovaj zadatak zahtijevao nabrajanja jer bi bilo potrebno uključiti sve mrežne segmente čiji se IP paketi mogu pojaviti uključujući i mrežni segment lokalnog računala. Jednostavnije, korištenjem negacije to je moguće izvesti naredbom

```
# iptables -s ! localhost ...
```

4.4. Određivanje protokola

Protokol se jednostavno može odrediti korištenjem parametra `-p` ili `--protocol` i navođenjem jednog od protokola ili po numeričkoj oznaci ili po nazivu pri čemu velika i mala slova imaju jednaku vrijednost. Primjerice, jedni od najčešće korištenih protokola su *TCP*, *UDP* i *ICMP*, a posebna oznaka *ALL* koristi se za određivanje svih protokola. Ukoliko se u okviru naredbe ovaj parametar ispusti, *ALL* je podrazumijevana vrijednost. Simbolička imena iz datoteke `/etc/protocols` također su dozvoljena kao i korištenje inverzije. Primjer korištenja je naredba u okviru koje se određuje protokol:

```
# iptables -p TCP ...
```

4.5. Sučelje

Kod sučelja se razlikuju dva logički neovisna oblika sučelja: ulazno i izlazno. Ulazno (eng. *input*) se određuje oznakama `-i` i `--in-interface`, a odlazno (eng. *output*) oznakama `-o` i `--out-interface`. Paket koji dolazi na *INPUT* lanac nema izlazno sučelje, kao što niti paket na *OUTPUT* lancu nema ulazno sučelje. Stoga je u kombinaciji s tim lancima onemogućeno kreiranje pravila koja uključuju ta sučelja, što se može uočiti na sljedećem primjeru.

```
# iptables -A INPUT -o eth1
iptables v1.3.7: Can't use -o with INPUT
```

Ponekad je poželjno odrediti više sličnih sučelja koja se dinamički mijenjaju i nisu uvijek sva prisutna na sustavu. Ovaj problem rješava se korištenjem oznake `+` koja predstavlja proizvoljan niz alfanumeričkih znakova. Na primjer, neka je potrebno odrediti sva *ethernet* sučelja (čije su inače oznake oblika *ethX*, gdje *X* označava redni broj sučelja). Pri tome, broj sučelja nije poznat, a varira u vremenu. Zadatak je najjednostavnije riješiti korištenjem spomenute oznake:

```
# iptables -i eth+ ...
```

I kod definiranja sučelja u pravilima je omogućeno korištenje inverzije.

4.6. Specifikacija fragmentiranih paketa

Razdijeljeni paketi (eng. *fragments*) koriste se kada je zbog veličine paketa nemoguće odaslati sve podatke u sklopu jednog paketa, nego ga je potrebno razdijeliti u više zasebnih paketa. Na odredištu se fragmentirani paketi spajaju u jedan i na taj način se rekonstruira čitav odaslani paket.
Problem koji se javlja kod ovakvih paketa je nemogućnost čitanja karakteristika izvornog paketa budući da je on upakiran u niz manjih paketa. Prema tome, čitanje zaglavlja koje određuje protokol paketa nije moguće budući da se ona nalaze samo u okviru prvog fragmenta. Koristi li se mehanizam praćenja veze ili NAT, svi fragmenti rekonstruirani su u izvorni paket prije nego dođu do mehanizma filtriranja pa u tom slučaju opisani problem nije prisutan.
Za rješavanje problema potrebno je razumjeti način na koji se pravila filtriranja primjenjuju na fragmentirane pakete. Jedino će prvi fragment ispravno dati odgovor na usporedbe s pravilima filtra dok svi ostali neće jer ne sadrže informacije zaglavlja izvornog paketa. Primjerice, korištenje pravila

```
# iptables -p TCP --sport www ...
```

kojim se određuju paketi pristigli od web poslužitelja, uspješno će se obaviti samo na prvom paketu.
Međutim, mehanizam IPTables vatrozida omogućuje i određivanje pravila posebno za drugog i ostale fragmente paketa upisivanjem opcije `-f` ili `--fragment` koja može biti i u kombinaciji s oznakom invertiranja `!` - pri čemu se određuje pravilo koje se ne primjenjuje na drugog i ostale fragmente. Primjer naredbe pravila kojim se odbacuju svi fragmenti s ciljnom adresom 192.168.1.122:

```
# iptables -A OUTPUT -f -d 192.168.1.1 -j DROP
```

4.7. Pravila vezana uz TCP protokol

Jedna od najčešće korištenih pravila vezano uz TCP protokol su provjera zastavica i priključaka (eng. *port*).
Opcija `--tcp-flags` koju slijede dva niza oznaka zastavica omogućava filtriranja temeljeno na TCP zastavicama. Prvi niz određuje provjeravane zastavice, a drugi niz označava one koje moraju biti postavljene. Primjerice, naredba

```
# iptables -A INPUT --protocol tcp --tcp-flags ALL SYN,ACK -j DROP
```

određuje pravilo koje zahtijeva provjeru svih zastavica oznakom *ALL*, ali jedino *SYN* i *ACK* trebaju biti postavljene. Ostale zastavice koje se mogu ispitivati su *FIN*, *RST*, *URG* i *PSH* te *NONE* koja označava paket bez postavljenih zastavica. Valja primijetiti da se u okviru jednog niza kod nabrajanja zastavica kao poveznik koristi zarez.
Opcija `--syn` koristi se kao kratica za izraz `--tcp-flags SYN,RST,ACK SYN`.
Svaki paket karakterizira izvorna i ciljna adresa uz izvorni i ciljni priključak (eng. *source/destination port*). Argument `--source-port`, ili kraće `--sport`, očekuje jedan priključak ili niz priključaka danih numerički ili simbolički, po imenu prema nazivima unutar `/etc/services` datoteke, te se provjerava u okviru izvornog priključka zapisanog u paketu. U kombinaciji s oznakom negacije može se koristiti za jednostavno određivanje niza priključaka. Posebna oznaka minus – uvedena je kao oznaka intervala, a može označavati interval priključaka omeđen navedenima, ali se jedan od njih može i ispustitit. U tom slučaju ispušteni priključak označava granični priključak. Primjerice, `--sport -123` označava niz priključaka od nultog do 123. priključka.
Ciljni priključak označava se parametrom `--destination-port` i `--dport`, a vrijedi ista konvencija kao za opisani izvorni priključak.
Oznaka `--tcp-option` služi za određivanje TCP oznake koju mora posjedovati paket da bi zadovoljio pravilo. Svaki paket koji nema potpuno TCP zaglavlje automatski biva odbačen pri pokušaju provjere TCP oznake.
Primjena pravila vezanih uz TCP zastavice može se ilustrirati sljedećim primjerom. Neka je potrebno omogućiti nekom lokalnom procesu uspostavu veze prema vanjskom *WWW* poslužitelju, ali ne dozvoliti vezu od poslužitelja prema istom ili nekom drugom lokalnom procesu. S obzirom da TCP komunikacija zahtijeva uspostavu ispravne veze (prosljeđivanje paketa u oba smjera), nije moguće samo blokirati

dolazne pakete. Pokazuje se da je ispravan način izvedbe blokiranje pokušaja uspostave veze kojeg je moguće odrediti na temelju postavljenih zastavica *SYN*, *ACK* i *FIN*. Odbacivanjem ovih paketa onemogućuje se uspostava veze u njezinom začetku.
Odgovarajući primjer koji specificira pokušaje uspostave veze s adrese 192.168.1.100 izgleda:

```
# iptables -p TCP -s 192.168.1.100 --syn ...
```

4.8. Pravila vezana uz UDP protokol

Ukoliko se koristi parametar `--protocol udp` automatski su omogućena pravila vezana uz određivanje izvornog i ciljnog priključka:

- `--source-port` ili kraće `--sport` i
- `--destination-port` odnosno `--dport`.

Način korištenja je identičan kao kod TCP protokola pa se neće zasebno opisivati.

4.9. Pravila vezana uz ICMP protokol

Određivanje kontrolnih paketa (eng. *ICMP - Internet Control Message Protocol*) izvodi se navođenjem parametara `--protocol icmp`. U ovom slučaju postoji samo jedan parametar, a to je određivanje vrste ICMP paketa. Njegov naziv je `--icmp-type` i zahtijeva navođenje odgovarajuće oznake. Odgovarajuća oznaka uključuje naziv tipa poruka, primjerice `host-unreachable` ili numeričke oznake `3` ili numeričke oznake tipa i koda odvojeno, npr. `3/3`. Popis postojećih naziva tipova paketa dohvaća se naredbom:

```
# iptables -p icmp --help
```

4.10. Pravila vezana uz NAT

U prethodnim cjelinama spomenuta je NAT funkcionalnost te su navedene temeljne značajke DNAT, SNAT i MASQUERADING mehanizama. Slijedi nekoliko primjera vezanih uz prevođenje adresa i priključaka.

```
# iptables -t nat -A POSTROUTING -p tcp -o eth0 -j SNAT --to-source
194.236.50.155-194.236.50.160:1024-32000
```

Opcija `--to-source` određuje kojima će se IP adresama i priključcima zamijeniti izvorne adrese i priključci paketa. Najjednostavnije korištenje obuhvaća upisivanje jedne IP adrese i opcionalno priključka odvojenog dvotočkom, npr. 127.0.0.1:8080. Niz IP adresa zapisuje se u obliku dviju rubnih adresa intervala povezanih povlakom –, npr. 192.168.1.100-192.168.1.200. Treba napomenuti da jedan tok paketa uvijek dobiva istu IP adresu. Isto pravilo vrijedi i za priključke.
Takva pravila može se koristiti samo u kombinaciji s `-p tcp` i `-p udp` opcijama.
DNAT mehanizam, iako u suštini bitno različit od SNAT mehanizma, ima identičnu sintaksu prethodno pojašnjenu i navodi se unutar *PREROUTING* lanca u *nat* tablici.
Akcija MASQUERADE razvijena je za slučajeve dinamičkog dodjeljivanja adresa sučeljima, zbog toga što SNAT u tom slučaju ne može zadovoljiti potrebe. Primjer korištenja opcije MASQUERADE dan je na sljedećem ispisu:

```
iptables -t nat -A POSTROUTING -p TCP -j MASQUERADE --to-ports 1024-31000
```

Opcija `--to-ports` koristi se za mijenjanje izvornih priključaka odlaznih paketa. Moguće je odrediti jedinstveni priključak, ali i niz priključaka od kojih se slučajnim odabirom uzima jedan. Kao što je uočljivo iz danog primjera, interval se određuje navođenjem rubnih priključaka odvojenih povlakom –. Pravilo je važeće samo ukoliko se odabere TCP ili UDP protokol parametrom `-p`.

4.11. Dodavanje, uklanjanje i mijenjanje zapisa iz tablica i lanaca

Nekoliko je načina na koji se može uređivati zapise u IPTables tablicama. U dosadašnjim primjerima dodavanje pravila u tablice koristilo se vrlo često, ali nije bilo opisano. Dodavanje naredbe na zadnju poziciju u lancu izvodi se navođenjem opcije `-A`, ukoliko se želi pisati u neku tablicu koja nije *filter* potrebno je i opcijom `-t` odrediti željenu tablicu, a zatim navesti pravilo. Primjer dodavanja naredbe u INPUT lanac:

```
# iptables -A INPUT -s 192.168.23.0/24 -j REJECT
```

Umetanje naredbe na željenu poziciju u lancu, obavlja se naredbom `-I` odnosno `--insert` uz određivanje pozicije umetanog pravila u tablici, kako slijedi u danom primjeru u kojem se naredba umeće na prvu poziciju:

```
# iptables -I INPUT 1 -s 192.168.23.34 -j ACCEPT
```

Zamjena postojećeg pravila s neke pozicije vrši se opcijom `-R` ili `--replace`:

```
# iptables -R INPUT 1 -s 192.168.23.34 -j DROP
```

Konačno, brisanje se izvodi navođenjem opcije `-D` ili `--delete` te unosom parametara pravila ili rednog broja pravila u tablici:

```
# iptables -D INPUT -s 192.168.23.0/24 -j REJECT
# iptables -D INPUT 1
```

Stvaranje novog lanca izvodi se korištenjem opcija `-N` ili `--new-chain` uz naziv novog lanca:

```
# iptables -N NoviLanac
```

Uklanjanje lanca se obavlja opcijama `-X` odnosno `--delete-chain`:

```
# iptables -X NoviLanac
```

Mijenjanje naziva lanaca može se izvesti korištenjem opcije `-E` ili `--rename-chain`:

```
# iptables -E StariNazivLanca NoviNazivLanca
```

Novo opće pravilo koje se primjenjuje nad kompletnom tablicom određuje se opcijom `-P` ili `--policy`:

```
# iptables -P INPUT REJECT
```

Uklanjanje svih pravila iz određenog lanca i tablice vrši se naredbom `-F` ili `--flush`. Ako se ne navede tablica, podrazumijevana tablica je *filter*. Ukoliko se ne navede lanac, brišu se sadržaji svih tablica u lancima u kojima se nalaze. Primjerice, naredba

```
# iptables -F -t nat
```

obrisat će sve zapise iz svih *nat* tablica, a one se nalaze u *PREROUTING* i *POSTROUTING* te *OUTPUT* lancima.
Ispis svih tablica sadržanih u odgovarajućim lancima dobiva se korištenjem opcije `-L` odnosno `--list`. Za odabir željene tablice potrebno je navesti i opciju -t, a podrazumijevana je *filter*. Ukoliko se doda i opcija `-v`, kao prvi stupac u tablici s ispisom dobiva se i broj zaprimljenih paketa koji su zadovoljili odgovarajuće pravilo.

```
# iptables -L -t nat -v
```

Slično vrijedi i kada se želi postaviti vrijednost brojača primijenjenih pravila na nulu korištenjem opcije `-Z` odnosno `--zero`:

```
# iptables -Z INPUT
```

4.12. Posebne usporedbe

Iz nekoliko prije navedenih primjera moglo se uočiti da se za neke naredbe koristi opcija `-m` odnosno `--match`, a tek onda opcija za usporedbu. Tu se radi o tzv. eksplicitnim usporedbama za koje je potrebno korištenje dodatnih programskih mehanizama. Između ostalih, na ovaj način se navode naredbe uspoređivanja stanja veze navođenjem opcije `-m state`. Ovakva funkcionalnost IPTables paketa omogućuje jednostavno dodavanje novih mehanizama provjere, a niz mogućnosti se proširuje svakom novom inačicom programskog paketa. Korisnost eksplicitnih usporedbi može biti velika, ako se za njima ukaže potreba, a prvenstveno ovisi o korisnikovom poznavanju alata i njegovim zahtjevima. Razlika između uobičajenih usporedbi i ovdje spomenutih je u tome što IPTables za uobičajene usporedbe automatski učitava potrebne module. U slučaju eksplicitnih usporedbi moduli se neće automatski učitati, nego se zahtijeva njihovo eksplicitno navođenje. Jedan od češće korištenih primjera je upotreba `iprange` usporedbe:

```
# iptables -A INPUT -p tcp -m iprange --src-range 192.168.1.13-
192.168.2.19
```

Ograničavanje paketa temeljeno na veličini može se izvesti naredbom

```
# iptables -A INPUT -p tcp -m length --length 1400:1500
```

što znači da se ovim pravilom pronalaze paketi s duljinom između 1400 i 1500 okteta.
Određivanje paketa temeljeno na adresi fizičkog sučelja (eng. *MAC - Ethernet Media Access Control*), izvodi se naredbom:

```
# iptables -A INPUT -m mac --mac-source 00:00:00:00:00:01
```

Sljedeća nerijetko korištena opcija omogućava obuhvaćanje više priključaka i zamjenjuje navođenje više uzastopnih `--source-port` ili `--destination-port` naredaba.

```
# iptables -A INPUT -p tcp -m multiport --source-port 22,53,80,110
# iptables -A INPUT -p tcp -m multiport --port 22,53,80,110
```

Oznaka `--port`, kako je prikazano drugom naredbom u primjeru, odnosi se i na ciljne i na izvorne priključke.
Filtriranje prema stanju veze izvodi se naredbom:

```
# iptables -A INPUT -m state --state RELATED,ESTABLISHED
```

Više puta već spominjana polja TOS i TTL mogu se ispitivati sljedećim pravilima:

```
# iptables -A INPUT -p tcp -m tos --tos 0x16
# iptables -A OUTPUT -m ttl --ttl-eq 60
```

5. Pomoćne naredbe za spremanje i obnovu konfiguracije

Paket IPTables isporučuje se uz nekoliko dodatnih alata kojima se olakšava korištenje vatrozida. Ovo je posebno korisno iz razloga što se sva pravila navedena u prethodnim poglavljima spremaju u tablice

u radnoj memoriji i po prestanku rada operacijskog sustava ista se gube. Očito, kod velikog niza pravila i različitih konfiguracija vatrozida ovi alati također nalaze svrhu.
Radi se o programima:

- `iptables-save` i
- `iptables-restore`.

Alat `iptables-save` koristi se za spremanje grupa pravila u posebno oblikovanu datoteku sa strogo definiranim sintaksnim pravilima koja ne odstupaju uvelike od uobičajenog zadavanja pravila korištenjem naredbenog retka.
Naredbu `iptables-restore` moguće je koristiti unutar skripta razvijenih proizvoljnim skriptnim jezicima. Jedini problem je što se tada prikupljeni podaci o pravilima trebaju usmjeriti na standardni ulaz (*stdin*) ovog procesa. Ovo je praktično i u slučaju stvaranja velikog niza pravila (reda veličine nekoliko tisuća) budući da je takav mehanizam prenošenja podataka vrlo brz. Primjer za to, uz pretpostavku da skripta `make_rules.sh` stvara niz pravila, izgleda:

```
# make_rules.sh | iptables-restore
```

Postavke `iptables` programa podrazumijevano se spremaju u `/etc/sysconfig/iptables` datoteku. Pored toga, valja napomenuti da korištenje simboličkih imena (npr. www.cert.hr) nije uputno budući da se servis `iptables` tijekom uključivanja sustava pokreće prije imenskog servisa (eng. *DNS – Domain Name System*) što znači da je onemogućeno pretvaranje simboličkih imena u IP adrese. Zato se savjetuje korištenje numeričkih IP adresa ukoliko je važno imati potpuno osposobljen vatrozid prije pokretanja drugih servisa.

5.1. Spremanje i čitanje pravila

Već je napomenuto da se sva pravila stvorena naredbom `iptables` spremaju u radnu memoriju. Ukoliko se sustav ponovno pokrene bez spremanja postojeće konfiguracije vatrozida, sva pravila su bespovratno izgubljena. Za spremanje pravila može se pokrenuti sljedeća naredba s `root` ovlastima:

```
$ /sbin/service iptables save
```

Ovom naredbom pokreće se izvođenje IPTables inicijalizacijske skripte koja pokreće program `/sbin/iptables-save` i zapisuje tekuću konfiguraciju u `/etc/sysconfig/iptables` datoteku. Postojeća `/etc/sysconfig/iptables` datoteka sprema se pod nazivom `/etc/sysconfig/iptables.save`. Pri sljedećem uključivanju sustava, inicijalizacijska skripta učitava pravila korištenjem `/sbin/iptables-restore` naredbe.
Spremanje konfiguracije IPTables paketa u neku proizvoljnu datoteku radi, primjerice, korištenja na drugom sustavu i slično, može se obaviti pokretanjem sljedeće naredbe:

```
# iptables-save > <ime_datoteke>
```

U slučaju prenošenja pravila na druga računala korištenjem `/etc/sysconfig/iptables` datoteke, za primjenu tih pravila potrebno je zaustaviti i ponovno pokrenuti IPTables servis naredbom:

```
# /sbin/service iptables restart
```

5.2. Skripte IPTables paketa

U prethodnom poglavlju opisano je korištenje nekoliko mogućih skripta za učitavanje i spremanje pravila te za zaustavljanje i ponovno pokretanje servisa. Korištenje inicijalizacijske skripte za upravljanje čitavim IPTables mehanizmom odvija se naredbom:

```
# /sbin/service iptables <naredba>
```

Slijedi opis mogućih vrijednosti parametar označen izrazom `<naredba>` uz obrazloženja.

- `start`
  Koristi se za pokretanje IPTables servisa pri čemu se svi pokrenuti IPTables servisi zaustavljaju, a pravila se učitavaju iz `/etc/sysconfig/iptables` datoteke korištenjem `/sbin/iptables-restore` naredbe.
- `stop`
  Zaustavlja pokrenuti vatrozid pri čemu se sva pravila iz memorije gube nepovratno ukoliko prethodno nisu spremljena. Ukoliko je parametar `IPTABLES_SAVE_ON_STOP` iz konfiguracijske datoteke `/etc/sysconfig/iptables-config` postavljen na `yes`, pravila se spremaju u `/etc/sysconfig/iptables` datoteku, a njezin prethodni sadržaj se sprema u `/etc/sysconfig/iptables.save` datoteku.
- `restart`
  Zaustavlja izvođenje vatrozida i pokreće ga ponovno. Uz vrijednost parametra `IPTABLES_SAVE_ON_RESTART` postavljenu na `yes`, tekuća pravila se spremaju u datoteku `/etc/sysconfig/iptables`, a njezin prethodni sadržaj se prepisuje u `/etc/sysconfig/iptables.save` datoteku.
- `status`
  Prikazuje stanje vatrozida i popis svih aktivnih pravila. Podrazumijevano, ovdje će se prikazati IP adrese unutar svakog pravila. Za prikaz simboličkog naziva u ovom slučaju, potrebno je unutar konfiguracijske datoteke postaviti vrijednost parametra `IPTABLES_STATUS_NUMERIC` na `no`.
- `panic`
  Naredba se koristi za brisanje svih postojećih pravila iz memorije pri čemu se opća pravila svih tablica postavljaju na DROP. Mogućnost je korisna u slučaju kada je sustav pod napadom. Postupak je elegantniji od isključivanja komunikacijskih kabela ili isključivanja sustava jer omogućuje forenzičku analizu trenutnog stanja.

6. Primjer korištenja IPTables naredbi

Radi jednostavnosti dan je primjer podešavanja IPTables vatrozida koji brani isključivo lokalno računalo. Sljedećim naredbama postavlja se temeljna funkcionalnost vatrozida, a moguće ih je integrirati unutar izvršne skripte. Na posljetku treba voditi računa o spremanju konfiguracije. Preporučivo je započeti čišćenjem postojećih zapisa i postavljanjem općih pravila lanaca:

```
iptables -P INPUT DROP
iptables -P FORWARD DROP
iptables –flush
iptables --delete-chain
```

Prve dvije naredbe postavljaju opće pravilo *INPUT* I *FORWARD* lanaca na odbacivanje paketa koji ne zadovoljavaju niti jedno pravilo iz odgovarajućih tablica. Ovaj postupak u skladu je s uvodnim napomenama o sigurnosti kod određivanja općih pravila.
Treća naredba koristi se za brisanje svih postojećih pravila, a potpuno je istovjetna brisanju pravila jedno po jedno. Važno je uočiti da se ovom naredbom ne utječe na opća pravila.
Posljednja naredba u nizu briše sve eventualne lance koje je korisnik stvorio, tako da ostaju samo ugrađeni lanci opisani u prethodnim cjelinama.
Slijedi niz koji postavlja minimalnu funkcionalnost vatrozida tako da se računalo može koristiti u uobičajene svrhe pristupa s Interneta.

```
iptables -A INPUT -i lo --source 127.0.0.1 --destination 127.0.0.1 -j
ACCEPT
```

Naredba dodaje (eng. *append*) novo pravilo koje određuje mogućnost slanja paketa sa sučelja 127.0.0.1 na isto sučelje.

```
iptables -A INPUT -m state --state "ESTABLISHED,RELATED" -j ACCEPT
```

Dodaje se pravilo *filter* tablici ulaznog lanca koje navodi kako se svi paketi iz veza sa stanjima ESTABLISHED ili RELATED prihvaćaju. Ovdje se koristi mehanizam praćenja veza.

```
iptables -A INPUT -p icmp --icmp-type destination-unreachable -j ACCEPT
iptables -A INPUT -p icmp --icmp-type time-exceeded -j ACCEPT
iptables -A INPUT -p icmp --icmp-type echo-request -j ACCEPT
iptables -A INPUT -p icmp --icmp-type echo-reply -j ACCEPT
```

Prethodne linije dozvoljavaju prolaz ICMP paketima odgovarajućih tipova.
Sljedeće naredbe dodaju pravila kojima se omogućava prihvaćanje TCP paketa namijenjenih *ssh*, *http* i *https* uslugama. Dakle, smatra se da lokalno računalo osigurava spomenute usluge te se ovim naredbama omogućuje njihov ispravan rad.

```
iptables -A INPUT -p tcp --dport ssh -j ACCEPT
iptables -A INPUT -p tcp --dport http -j ACCEPT
iptables -A INPUT -p tcp --dport https -j ACCEPT
```

Konačno, spremanje postojećeg stanja vatrozida moguće je izvesti sljedećom naredbom:

```
iptables-save > /etc/sysconfig/iptables
```

Jednostavnim pokretanjem skripte sastavljene od prethodnih naredbi, uz dodavanje odgovarajućeg zaglavlja na sam početak datoteke

```
#!/bin/sh
```

moguće je podesiti vrlo jednostavan niz pravila IPTables mehanizma. Potom je potrebno promijeniti atribute skriptnoj datoteci i pokrenuti ju, kako je prikazano primjerom:

```
# chmod u+x firewall.sh
# firewall.sh
```

Primjer još jedne jednostavne, a potpuno funkcionalne skripte s opisima u okviru iste dan je na sljedećem ispisu.

```
#!/bin/sh

# Omogući nesmetan promet na povratnom sučelju
/sbin/iptables -A INPUT -i lo -j ACCEPT
/sbin/iptables -A OUTPUT -o lo -j ACCEPT

# Postavljanje podrazumijevanih vrijednosti općih pravila
/sbin/iptables --policy INPUT DROP
/sbin/iptables --policy OUTPUT DROP
/sbin/iptables --policy FORWARD DROP

# Dopusti neograničen izlazni promet
/sbin/iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
/sbin/iptables -A OUTPUT -m state --state NEW,ESTABLISHED,RELATED -j
ACCEPT

# Omogući ulazni promet preko priključka 22 (SSH) računala s
# adresom 192.168.1.100
/sbin/iptables -A INPUT -p tcp -s 192.168.1.100 --dport 22 -m state --
state NEW -j ACCEPT

# Odbaci sav ostali promet
/sbin/iptables -A INPUT -j DROP
```

```
# Ako se želi zabilježiti odbačene pakete, prethodna naredba može se
# zamijeniti sljedećim nizom (ukloniti komentare)
# Stvori novi lanac koji će preusmjeravat pakete na logiranje pa
# zatim na odbacivanje
#   /sbin/iptables -N LOGDROP
#   /sbin/iptables -A LOGDROP -j LOG
#   /sbin/iptables -A LOGDROP -j DROP
# Odbaci sav ostali promet uz logiranje
#   /sbin/iptables -A INPUT -j LOGDROP

# Osiguraj primjenu ovih pravila kod pokretanja IPTables servisa
/sbin/service iptables save
```

Slijedi primjer nešto naprednije skripte uz odgovarajuća obrazloženja.

```
#!/bin/sh

# Podešavanja postavki vezanih uz Internet.
INET_IP="194.236.50.155"
INET_IFACE="eth0"
INET_BROADCAST="194.236.50.255"

# Podešavanje postavki vezanih uz lokalnu mrežu.
LAN_IP="192.168.0.2"
LAN_IP_RANGE="192.168.0.0/16"
LAN_IFACE="eth1"

# Podešavanje sučelja lokalnog računala.
LO_IFACE="lo"
LO_IP="127.0.0.1"

# Makro za jednostavnije pokretanje iptables naredbe
IPTABLES="/usr/sbin/iptables"

# Najprije je potrebno inicijalno pronaći međuzavisnosti modula.
/sbin/depmod -a

# Učitati u jezgru potrebne module.
/sbin/modprobe ip_tables
/sbin/modprobe ip_conntrack
/sbin/modprobe iptable_filter
/sbin/modprobe iptable_mangle
/sbin/modprobe iptable_nat
/sbin/modprobe ipt_LOG
/sbin/modprobe ipt_limit
/sbin/modprobe ipt_state

# Popis preostalih modula
#/sbin/modprobe ipt_owner
#/sbin/modprobe ipt_REJECT
#/sbin/modprobe ipt_MASQUERADE
#/sbin/modprobe ip_conntrack_ftp
#/sbin/modprobe ip_conntrack_irc
#/sbin/modprobe ip_nat_ftp
#/sbin/modprobe ip_nat_irc

# Podešavanje /proc mehanizma
echo "1" > /proc/sys/net/ipv4/ip_forward

# Ostale mogućnost podešavanja /proc mehanizma
#echo "1" > /proc/sys/net/ipv4/conf/all/rp_filter
#echo "1" > /proc/sys/net/ipv4/conf/all/proxy_arp
#echo "1" > /proc/sys/net/ipv4/ip_dynaddr
```

```
# Postavljanje općih pravila za filter tablicu
$IPTABLES -P INPUT DROP
$IPTABLES -P OUTPUT DROP
$IPTABLES -P FORWARD DROP

Stvaranje lanca za neispravne TCP pakete
$IPTABLES -N bad_tcp_packets

# Stvaranje lanaca za ICMP, TCP i UDP pakete
$IPTABLES -N allowed
$IPTABLES -N tcp_packets
$IPTABLES -N udp_packets
$IPTABLES -N icmp_packets

# Postavljanje pravila u prethodno stvorene lance
$IPTABLES -A bad_tcp_packets -p tcp --tcp-flags SYN,ACK SYN,ACK \
-m state --state NEW -j REJECT --reject-with tcp-reset
$IPTABLES -A bad_tcp_packets -p tcp ! --syn -m state --state NEW -j LOG \
--log-prefix "New not syn:"
$IPTABLES -A bad_tcp_packets -p tcp ! --syn -m state --state NEW -j DROP

$IPTABLES -A allowed -p TCP --syn -j ACCEPT
$IPTABLES -A allowed -p TCP -m state --state ESTABLISHED,RELATED -j ACCEPT
$IPTABLES -A allowed -p TCP -j DROP

# TCP pravila
$IPTABLES -A tcp_packets -p TCP -s 0/0 --dport 21 -j allowed
$IPTABLES -A tcp_packets -p TCP -s 0/0 --dport 22 -j allowed
$IPTABLES -A tcp_packets -p TCP -s 0/0 --dport 80 -j allowed
$IPTABLES -A tcp_packets -p TCP -s 0/0 --dport 113 -j allowed

# UDP pravila
#$IPTABLES -A udp_packets -p UDP -s 0/0 --destination-port 53 -j ACCEPT
#$IPTABLES -A udp_packets -p UDP -s 0/0 --destination-port 123 -j ACCEPT
#$IPTABLES -A udp_packets -p UDP -s 0/0 --destination-port 2074 -j ACCEPT
#$IPTABLES -A udp_packets -p UDP -s 0/0 --destination-port 4000 -j ACCEPT

# Onemogućavanje logiranja broadcast paketa
#$IPTABLES -A udp_packets -p UDP -i $INET_IFACE -d $INET_BROADCAST \
#--destination-port 135:139 -j DROP

# Onemoućavanje logiranja DHCP prometa
#$IPTABLES -A udp_packets -p UDP -i $INET_IFACE -d 255.255.255.255 \
#--destination-port 67:68 -j DROP

# ICMP pravila
$IPTABLES -A icmp_packets -p ICMP -s 0/0 --icmp-type 8 -j ACCEPT
$IPTABLES -A icmp_packets -p ICMP -s 0/0 --icmp-type 11 -j ACCEPT

# Pravilo za neispravne i neželjene TCP pakete
$IPTABLES -A INPUT -p tcp -j bad_tcp_packets

# Pravila za mreže koje nisu dio Interneta
$IPTABLES -A INPUT -p ALL -i $LAN_IFACE -s $LAN_IP_RANGE -j ACCEPT
$IPTABLES -A INPUT -p ALL -i $LO_IFACE -s $LO_IP -j ACCEPT
$IPTABLES -A INPUT -p ALL -i $LO_IFACE -s $LAN_IP -j ACCEPT
$IPTABLES -A INPUT -p ALL -i $LO_IFACE -s $INET_IP -j ACCEPT

# Pravilo za nepravilne DHCP zahtjeve s lokalne mreže
$IPTABLES -A INPUT -p UDP -i $LAN_IFACE --dport 67 --sport 68 -j ACCEPT

# Pravila za dolazeće pakete s Interneta
$IPTABLES -A INPUT -p ALL -d $INET_IP -m state --state ESTABLISHED,RELATED
\
```

```
-j ACCEPT
$IPTABLES -A INPUT -p TCP -i $INET_IFACE -j tcp_packets
$IPTABLES -A INPUT -p UDP -i $INET_IFACE -j udp_packets
$IPTABLES -A INPUT -p ICMP -i $INET_IFACE -j icmp_packets

# Za izbjegavanje ispunjavanja dnevnika (log), onemogućiti logiranje
# multicast paketa
#$IPTABLES -A INPUT -i $INET_IFACE -d 224.0.0.0/8 -j DROP

# Zabilježi sumnjive pakete u dnevnik
$IPTABLES -A INPUT -m limit --limit 3/minute --limit-burst 3 -j LOG \
--log-level DEBUG --log-prefix "IPT INPUT packet died: "

# #### FORWARD lanac ####
# Određivanje neželjenih TCP paketa u FORWARD lancu
$IPTABLES -A FORWARD -p tcp -j bad_tcp_packets

# Prihvaćanje paketa koji se zaista i trebaju proslijediti
$IPTABLES -A FORWARD -i $LAN_IFACE -j ACCEPT
$IPTABLES -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT

# Logirati neispravne pakete koji ne zadovoljavaju prethodna pravila
$IPTABLES -A FORWARD -m limit --limit 3/minute --limit-burst 3 -j LOG \
--log-level DEBUG --log-prefix "IPT FORWARD packet died: "

# #### OUTPUT lanac ####
# Nepoželjni paketi u OUTPUT lancu
$IPTABLES -A OUTPUT -p tcp -j bad_tcp_packets

# Posebna pravila za OUTPUT lanac koja određuju dozvoljene IP adrese
$IPTABLES -A OUTPUT -p ALL -s $LO_IP -j ACCEPT
$IPTABLES -A OUTPUT -p ALL -s $LAN_IP -j ACCEPT
$IPTABLES -A OUTPUT -p ALL -s $INET_IP -j ACCEPT

# Logirati pakete koji ne zadovoljavaju prethodna pravila
$IPTABLES -A OUTPUT -m limit --limit 3/minute --limit-burst 3 -j LOG \
--log-level DEBUG --log-prefix "IPT OUTPUT packet died: "

# Omogući prosljeđivanje IP paketa i NAT
$IPTABLES -t nat -A POSTROUTING -o $INET_IFACE -j SNAT --to-source
$INET_IP
```

Za daljnja podešavanja i proučavanja preporuča se korištenje literature navedene u popisu referenci, a savjetuje se i analiziranje dostupnih skripta kojima se stvaraju IPTables pravila. Jedna od korisnijih web stranica je i http://easyfwgen.morizot.net/gen/. Ovdje se radi o generatoru skriptnog koda koji osigurava postavljanje iptables vatrozida za jedno računalo, te uz mnoštvo komentara omogućuje razumijevanje korištenih postavki.

7. Zaključak

Problematika opisana u okviru ovog dokumenta sugerira da IPTables vatrozid predstavlja složeni mehanizam. Upoznavanje svih mogućnost vatrozida velik je i zahtjevan posao. Radi se o vrlo kvalitetnom mehanizmu koji, između ostalog, implementira statičko i dinamičko filtriranje. S obzirom na zahtjeve dinamičkog filtriranja za resursima, IPTables je realiziran optimalno. Međutim, postoje i čimbenici na koje korisnik vatrozida može utjecati postavljanjem odgovarajućih pravila koji također utječu na performanse vatrozida tako da nije samo programska izvedba presudna za kvalitetan rad alata.

Budući da je u dokumentu dan tek pregled značajnijih mogućnost, očigledno je kako se radi o vrlo moćnom alatu. Velik broj podržanih mogućnosti opravdanje je složenosti mehanizma. Pokriven je velik raspon opcija, počevši od prevođenja adresa, preko dinamičkog praćenja veza pa sve do modifikacije odgovarajućih polja mrežnih paketa. Jedna od prednosti je i mogućnost zaštite na više slojeva TCP stoga, što znači da je vatrozid u mogućnosti analizirati zaglavlja niske razine (poput dohvaćanja MAC adresa), ali i izvoditi analizu na višim razinama (TCP i UDP) pa sve do visoke razine, odnosno aplikacijskog sloja (FTP, IRC, ICQ kod praćenja veze).

Proširenja se lako implementiraju zahvaljujući činjenici da je alat razvijen s podrškom za modularnost. To ujedno i znači da se svakom novom inačicom paketa dodaju nove mogućnosti i podiže razina zaštite koju omogućuje vatrozid. Zahvaljujući implementiranom mehanizmu prevođenja adresa i načinu njegove izvedbe, omogućeno je i implementiranje jednolikog raspoređivanja opterećenja (eng. *load balancing*) pravilnim raspoređivanjem prosljeđivanih paketa.

Negativna strana zasigurno je potreba za ovladavanjem alata, budući da se radi o velikim količinama informacija. Jedan od nezgodnijih zahtjeva je i modifikacija pravila. Naime, u kompleksnijim sustavima s velikim brojem različitih zahtjeva, velik je i broj pravila. Iako će se sami mehanizam jako dobro nositi s takvim opterećenjem, administrator sustava zasigurno će imati velikih poteškoća kod izmjena. Zato se pribjegava različitim generatorima pravila, pisanjima skripti i dr.

Konačan zaključak je taj da se radi o jako dobrom mehanizmu čija kompleksnost je ujedno i prednost, ali i mana, pogotovo za manje zahtjevne korisnike.

Za daljnja razmatranja i saznanja, korisnike se potiče na konzultiranje navedenih referenci, ali u prvom redu na proučavanje tzv. *manual pages* opisa korištenja IPTables alata do kojeg je moguće doći iz naredbenog retka korištenjem naredbe:

```
# man iptables
```

8. Reference

[1.] The netfilter.org "iptables" project, http://www.netfilter.org/projects/iptables/index.html, srpanj 2007.

[2.] Stjepan Groš: „Mreže računala – pripreme za laboratorijske vježbe“, 2003.

[3.] The netfilter.org project, http://www.netfilter.org/, srpanj 2007.

[4.] IPTables tutorial, http://iptables-tutorial.frozentux.net/iptables-tutorial.html, srpanj 2007.

[5.] Netfilter/iptables Wikipedia, http://en.wikipedia.org/wiki/Iptables, srpanj 2007.

[6.] Linux iptables HOWTO, http://www.linuxguruz.com/iptables/howto/, srpanj 2007.

[7.] iptables(8) - Linux man page, http://linux.die.net/man/8/iptables, srpanj 2007.

[8.] Man page of IPTABLES, http://iptables-tutorial.frozentux.net/other/iptables.html, srpanj 2007.

[9.] IPTables – RedHat documentation, http://www.redhat.com/docs/manuals/enterprise/RHEL-5-manual/Deployment_Guide-en-US/ch-iptables.html, srpanj 2007.

[10.] HOWTO Iptables and stateful firewalls, http://gentoo-wiki.com/HOWTO_Iptables_and_stateful_firewalls, srpanj 2007.

[11.] Learning iptables from scratch, http://www.iptableslinux.com/learning-iptables-from-scratch.htm, srpanj 2007.

[12.] IPTables, http://defindit.com/readme_files/iptables.html, srpanj 2007.

[13.] IPTables Basics, http://www.justlinux.com/nhf/Security/IPtables_Basics.html, srpanj 2007.