HRVATSKA AKADEMSKA I ISTRAŽIVAČKA MREŽA
CROATIAN ACADEMIC AND RESEARCH NETWORK

PF vatrozid

CCERT-PUBDOC-2006-12-178

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

Razvojem računala i mrežne opreme, dostupnost Internetu i općenito mrežnim resursima, porasla je u velikoj mjeri. Nažalost, s rastom svjetske računalne mreže porastao je i broj zlonamjernih korisnika s različitim motivima i ciljevima napada. Zbog sve većeg broja zlonamjernih korisnika, kao i zbog sve većeg broja zlonamjernih programa (računalni virusi, crvi, trojanski konji i sl.), organizacije su prisiljene razvijati različite metode obrane svojih računalnih mreža. Aplikacije poput vatrozida (eng. *firewall*) koje implementiraju filtriranje mrežnih paketa (eng. *packet filter*) postaju neophodne za ispravan odnosno siguran rad svih računalnih mreža.

Packet Filter (PF) je programski vatrozid namijenjen BSD operacijskim sustavima, a koristi se za filtriranje TCP/IP prometa i za pretvaranje IP adresa (eng. NAT - *Network Address Translation*). Pored navedenih karakteristika, temeljnih svakom vatrozidu, PF omogućuje određivanje prioriteta paketima korištenjem ALTQ (eng. *Alternate Queueing*) programske podrške – kvalitetu usluge (eng. QoS – *Quality of Service*) i određivanje broja propuštenih paketa u jedinici vremena – propusnost (eng. *bandwidth*). Jedna od mogućnosti vatrozida je spremanje različitih zapisa o prometu u obliku dnevničkih zapisa (eng. *logging*).

Packet Filter je dio jezgre OpenBSD operacijskih sustava. Autor vatrozida je Daniel Hartmeier, a program je razvijan tijekom ljeta i jeseni 2001. godine i to ubrzanim tempom. Razlog tomu su nesuglasice s licencom prethodnog vatrozida, naziva IPFilter, čiji autor nije dozvoljavao izmjene unutar izvornog kôda. Stoga je taj paket odbačen i nekoliko tjedana postojala je distribucija OpenBSD sustava bez ikakvog ugrađenog vatrozida.

U dokumentu je opisan način korištenja PF vatrozida pri čemu su opisani postupci osnovne i naprednije konfiguracije. Na kraju je dan i primjer konfiguracije za malenu računalnu mrežu.

2. Osnove PF vatrozida

PF je od osnovnog vatrozida koji je podržavao samo filtriranje mrežnih paketa temeljeno na adresama, portovima, protokolu i vrstama paketa, prerastao u aplikaciju s puno većim stupnjem složenosti. Filtriranje trenutno podržava i prepoznavanje operacijskih sustava s kojih su paketi poslani. Pored očekivane funkcionalnosti, implementiran je i ALTQ (eng. *Alternate Queueing*) mehanizam koji je sada ugrađen u PF paket te omogućuje oblikovanje mrežnog prometa ograničavanjem propusnosti i određivanjem prioriteta nad paketima odnosno vezama.

Podešavanje PF vatrozida omogućeno je postavkama unutar jedne jedine konfiguracijske datoteke, obično nazvane `pf.conf` koja se nalazi u `/etc` direktoriju. Inačice ovog paketa isporučuju se zajedno s OpenBSD, FreeBSD, DragonFlyBSD i NetBSD operacijskim sustavima.

PF radi u načinu dinamičkog filtriranja paketa (eng. *stateful inspection*, *dynamic packet filtering*). Za razliku od statičkog filtriranja paketa kada se provjeravaju jedino parametri iz zaglavlja paketa, u dinamičkom načinu rada provjeravaju se zapisi o svim uspostavljenim vezama i ukoliko je analizirani paket povezan s nekom od tih uspostavljenih veza, na njega se ne primjenjuje filtar. Pored kontroliranja zaglavlja, mogu se provjeravati i sadržaji paketa raspakiravanjem sve do aplikacijskog sloja, ali ovo nije prvenstvena funkcionalnost aplikacije niti je u potpunosti implementirana. Riječ je samo o mogućnosti prepoznavanja nekih napada analizom paketa na višoj razini. Podaci o stanju uspostavljenih veza zapisuju se u odgovarajuće tablice stanja. Stoga donošenje odluka nije temeljeno samo na pravilima koja postavlja administrator nego je i ovisno o stanju u kojem se veza nalazi, a ono je pak određeno na temelju prethodno propuštenih ili blokiranih paketa. Dodatna sigurnosna odlika *stateful inspection* načina filtriranja paketa je zatvaranje svih portova sve dok veza ne zatraži eksplicitno otvaranje određenog porta nakon čega se provjerava da li se isti nalazi u listi dozvoljenih portova za korištenje. Ovo je sigurnosna mjera protiv pregledavanja portova (eng. *port scanning*), napada kojima napadači analiziraju otvorene portove na računalu.

Vrlo korisna funkcionalnost je i logiranje mrežnog prometa pri čemu dobiveni podaci mogu poslužiti za daljnje analize te eventualne detekcije napada, otkrivanja napadača i sl.

2.1. Razlike Linux i BSD operacijskih sustava

Za pravilno konfiguriranje PF programskog paketa potrebno je poznavati temelje BSD (eng. *Berkeley Software Distribution*) operacijskih sustava, a budući da je velik broj korisnika redovno bolje upoznat s Linux sustavima, u ovom poglavlju opisane su glavne razlike.

Mrežna sučelja na Linux sustavima označena su kao `eth0`, `eth1`, itd. Na BSD sustavima nazivi potječu od naziva pogonskih aplikacija i odgovarajućeg rednog broja počevši od nule. Primjerice, za Intelove mrežne kartice pogonske aplikacije označavaju se oznakom `em` što uz redni broj daje oznake `em0`, `em1`, itd., za kartice tvrtke SMC oznake su `sm0`, `sm1`, a ista logika vrijedi i za uređaje ostalih proizvođača.

Općenito BSD sustavi koriste `/etc/rc.conf` datoteku koju čita `/etc/rc` skripta i izvodi naredbe iz nje. Na OpenBSD sustavu preporuča se korištenje `/etc/rc.conf.local` datoteke za lokalna podešavanja budući da `rc.conf` sadrži tvorničke postavke. FreeBSD preferira zadržavanje podrazumijevanih vrijednosti unutar `/etc/defaults/rc.conf` datoteke pa je prema tome `/etc/rc.conf` odgovarajuća datoteka za obavljanje izmjena.

Konfiguracijska datoteka `pf.conf` koristi se u kombinaciji s `pfctl` naredbom koja posjeduje velik broj opcija. Postoje i web sučelja za administrativne poslove, ali ne isporučuju se kao standardni dio operacijskog sustava. Pored toga, nije razvijeno niti jedno grafičko sučelje koje je nedvojbeno jednostavnije za korištenje od kombiniranja uređivača tekstova, `pf.conf` konfiguracijske datoteke i `pfctl` naredbe.

3. Konfiguracija PF vatrozida i operacijskog sustava

Kao što je već spomenuto u prethodnom tekstu, konfiguracija vatrozida ovisi o operacijskom sustavu. Zato su u ovom poglavlju opisane konfiguracije za tri inačice BSD operacijskih sustava: OpenBSD, FreeBSD i NetBSD.

3.1. OpenBSD

U svrhu jednostavnog opisa, u nastavku ovog poglavlja pretpostavljena je situacija s nekoliko računala u lokalnoj mreži s tim da je na jednom od njih uključen PF vatrozid. Isto to računalo je preko svog drugog mrežnog sučelja spojeno na Internet te ono ima funkciju povezivanja ostalih računala s Internetom. Unutar `/etc/rc.conf.local` datoteke potrebno je postaviti liniju:

```
pf=YES
```

koja uključuje automatsko pokretanje PF servisa prilikom pokretanja operacijskog sustava. Dodatno, moguće je postaviti sljedeću liniju koja pokazuje PF paketu apsolutni put do konfiguracijske datoteke:

```
pf_rules=/etc/pf.conf
```

Prilikom sljedećeg pokretanja sustava, PF će automatski biti pokrenut i pri tome će čitati konfiguracijske podatke iz navedene datoteke. Inicijalno su sve postavke unutar te datoteke komentirane pa ih je potrebno korigirati. No, prije toga može se pokrenuti PF korištenjem sljedećih dviju naredbi pod pretpostavkom da korisnik nije prijavljen na sustav kao *root* korisnik (u protivnom naredba `sudo` nije potrebna):

```
sudo pfctl -e
sudo pfctl -f /etc/pf.conf
```

Prva naredba omogućuje uslugu filtriranja paketa, a druga pokazuje na datoteku s pravilima. U slučaju nepravilne konfiguracijske datoteke ili pojave sličnog problema koji bi onemogućio učitavanje pravila, postoji temeljni skup pravila koji se učitava tokom pokretanja sustava i to prije nego su mrežna sučelja uključena. Na taj način omogućuje se SSH pristup s proizvoljne izvorne adrese, dohvat adrese računala (eng. *Basic Host Resolution*) i montiranje NFS diskova.
Postavljanje konfiguracijske datoteke neovisno je o operacijskom sustavu pa će biti opisano nakon opisa podešavanja ostalih sustava.

3.2. FreeBSD

Postavke za automatsko pokretanje vatrozida na FreeBSD operacijskim sustavima zahtijevaju malo više napora nego kod OpenBSD sustava. Unutar datoteke `/etc/rc.conf` potrebno je dodati:

```
pf_enable="YES"
pf_rules="/etc/pf.conf"
pf_flags=""
pflog_enable="YES"
pflog_logfile="/var/log/pflog"
pflog_flags=""
```

Ukoliko je vatrozid namijenjen čitavoj računalnoj mreži, tada je potrebno dodati i liniju koja omogućuje prosljeđivanje paketa (eng. *Packet forwarding*):

```
gateway_enable="YES"
```

PF je uključen u FreeBSD od inačice 5.3 pa nadalje i to kao modul jezgre sustava (eng. *Kernel loadable module*) što znači da njegovo učitavanje treba pokrenuti naredbom:

```
sudo kldload pf
```

Omogućavanje (eng. *enable*) i onemogućavanje (eng. *disable*) filtriranja poruka zahtijeva pokretanje narednih naredbi:

```
pfctl -e
pfctl -d
```

Ukoliko je `rc.conf` datoteka podešena kako je prethodno pojašnjeno, PF se može pokrenuti naredbom:

```
sudo /etc/rc.d/pf start
```

3.3. NetBSD

Na NetBSD operacijskim sustavima inačica 2.0 i novijima, PF je dostupan kao modul jezgre sustava kao što je slučaj i kod FreeBSD operacijskih sustava. Instalira se korištenjem paketa `pkgsrc/security/pflkm` ili se prevede tako da postane statički dio jezgre.
Za postavljanje PF paketa unutar jezgre sustava, potrebno je u konfiguracijsku datoteku jezgre dodati sljedeće linije:

```
pseudo-device pf
pseudo-device pflog
```

Unutar `/etc/rc.conf` datoteke potrebno je dodati linije:

```
lkm="YES"
pf="YES"
pflogd="YES"
```

Nakon instalacije, modul je moguće učitati naredbama:

```
sudo modload /usr/lkm/pf.o
sudo pfctl -e
```

Alternativa tomu je pokretanje `rc` skripti na sljedeći način:

```
sudo /etc/rc.d/pf start
sudo /etc/rc.d/pflogd start
```

Za automatsko učitavanje modula tokom pokretanja sustava treba dodati sljedeću liniju u `/etc/lkm.conf` datoteku:

```
/usr/lkm/pf.o - - - - AFTERMOUNT
```

4. Izrada pravila

Nakon inicijalnog podešavanja u ovisnosti o operacijskom sustavu, može se prijeći na korak izrade jednostavnog skupa pravila. Konfiguracijska datoteka ima sintaksu neovisnu o samom sustavu pa je procedura opisana u poglavljima koja slijede jedinstvena, odnosno vrijedi za svaki BSD operacijski sustav.

4.1. Odsječci u konfiguracijskoj datoteci

Radi lakšeg snalaženja, konfiguracijsku datoteku se može podijeliti na nekoliko logički odvojenih, ali nesamostalnih cjelina. S izuzetkom makro naredbi i tablica, cjeline se trebaju pojaviti u konfiguracijskoj datoteci istim redoslijedom kojim su navedene u popisu. Prazne linije se ignoriraju, a komentari se označavaju znakom # na početku linije. Preporučeni dijelovi su:

- makro naredbe (eng. *macros*): korisničke varijable koje mogu čuvati IP adrese, imena mrežnih sučelja i slično,
- tablice (eng. *tables*): strukture koje čuvaju listu IP adresa,

- opcije (eng. *options*): razne opcije kojima se podešava rad PF vatrozida,
- *scrub*: obrada paketa radi normalizacije i defragmentacije,
- postavljanje u red (eng. *queueing*): kontrola propusnosti i određivanje prioriteta paketima,
- prevođenje (eng. *translation*): kontrola NAT usluge i usmjeravanja paketa,
- pravila filtriranja (eng. *filter rules*): omogućava selektivno filtriranje ili blokiranje paketa.

Primjeri entiteta navedeni su u nastavku dokumenta kroz primjere ostvarivanja različitih funkcionalnosti.

4.2. Podešavanje osnovne funkcionalnosti

Radi jasnoće pretpostavlja se jedno računalo na kojem nije pokrenut niti jedan servis, a ono komunicira samo s jednom mrežom koja može biti i Internet. Unutar `/etc/pf.conf` datoteke potrebno je upisati linije:

```
block in all
pass out all keep state
```

Prva linija onemogućuje prolaz ulaznog mrežnog prometa u sustav. Drugom linijom je omogućen izlaz svih paketa iz sustava uz zadržavanje informacija o stanju veza što omogućava *stateful inpection* funkcionalnost filtriranja paketa. To omogućava uspostavu funkcionalnih veza, odnosno propuštanje paketa unutar sustava, ali samo za veze koje su inicijalno stvorene od strane sustava. U protivnom, uspostava veze ne bi bila moguća.
Ukoliko se inicijalno zabrani ulaz za sav promet, kasnije je potrebno dodati pravila koja dozvoljavaju ulazak pojedinih vrsta paketa. U postavkama je moguće koristiti i drugačiji pristup. Inicijalno se dodijeli dozvola propuštanja svih paketa u sustav, a zatim pojedinačnim pravilima zabranjuje određenim vrstama paketa prolazak u sustav. Prvi pristup je nešto sigurniji jer ukoliko administrator zaboravi postaviti određena pravila koja dozvoljavaju određeni oblik prometa, nivo sigurnosti ostaje isti te eventualno prijeti određena nefunkcionalnost sustava. U drugom slučaju, ako administrator propusti postaviti određeno pravilo zabrane, sustav bi mogao postati ranjiv.

4.3. Određivanje i pregled sigurnosnih pravila

Preporučena globalna politika je onemogućiti propuštanje i ulaznog i izlaznog mrežnog prometa. Naredba koja implementira tu funkcionalnost izgleda ovako:

```
block all
```

Radi jednostavnijeg podešavanja, definiraju se dvije makro naredbe vezane uz vrste veza koje se koriste, TCP (eng. *Transmission Control Protocol*) ili UDP (eng. *User Datagram Protocol*).

```
tcp_services="{ ssh, smtp, domain, www, pop3, auth, pop3s }"
udp_services="{ domain }"
```

Prethodne linije pokazuju kako PF razumije definiranje portova korištenjem simboličkih naziva, a određivanje istih korištenjem brojeva je podrazumijevano. Popis svih portova i njihovih imena dan je u `/etc/services` datoteci. Potrebno je uočiti kako makro naredbe dozvoljavaju unose u obliku liste što pojednostavljuje i omogućuje veću preglednost pravila.
Nakon unosa prethodna dva retka u `pf.conf` datoteku, potrebno je dodati i konkretna pravila uključujući i pravilo o inicijalnom blokiranju kompletnog prometa.

```
block all
pass out proto tcp to any port $tcp_services keep state
pass proto udp to any port $udp_services keep state
```

Kao što je već spomenuto, `keep state` omogućava *stateful inspection* funkcionalnost filtriranja mrežnih paketa. Svaka veza koju sustav uspostavi, i to TCP ili UDP protokolom na odgovarajućim

portovima, uspješno bi prosljeđivala svoje pakete prema van, ali bez navedenog izraza povratni paketi bi bili odbačeni tj. ne bi im se dopustio ulazak u sustav. Zato PF čuva podatke o svim vezama kako bi omogućio dolazak povratnih paketa u sustav, i to samo onih paketa koji su dio prethodno uspostavljene veze. I da bi se ta funkcionalnost omogućila potrebo ju je eksplicitno uključiti izrazom `keep state`. U danom primjeru se i za UDP vrstu veze koristio isti izraz, iako on nema previše smisla budući da se radi o protokolu koji ne uspostavlja vezu kao što to čini TCP protokol koji u tu svrhu koristi različite zastavice (eng. *flags*). UDP paketi ne posjeduje te zastavice koje omogućuju uspostavljanje veze kroz različita stanja pa se za UDP protokol koristi tzv. *stateless* funkcionalnost filtriranja mrežnih paketa. Bez obzira na karakteristike UDP protokola, PF uspijeva sačuvati podatke o vezi budući da bi u protivnom, primjerice, DNS usluga bila onemogućena. Način na koji to čini je jednostavno čuvanje stanja određeni vremenski period od trenutka kada je odgovarajući paket propušten. Nakon isteka vremena, stanje se odbacuje i procedura kreće iznova.
Pokretanjem sljedeće naredbe primjenjuju se prethodno definirana pravila:

```
sudo pfctl -f /etc/pf.conf
```

Ukoliko nema sintaksnih pogrešaka, izvođenje dane naredbe neće uzrokovati nikakav ispis. U protivnom bi ispis mogao biti poput sljedećeg koji prikazuje pojavu sintaksne pogreške:

```
FreeBSD# pfctl -f /etc/pf.conf
/etc/pf.conf:91: syntax error
pfctl: Syntax error in config file: pf rules not loaded
```

Ako se pak ukaže potreba za detaljnijim ispisom, moguće je naredbi dodati i parametar `-v`.
Kod većih izmjena ili dopuna pravila poželjno je prije primjene provesti sintaksnu provjeru i to na sljedeći način pri čemu oznaka `-n` označava kako definirana pravila nije potrebno pokrenuti nego samo parsirati, a oznaka `-f` označava datoteku:

```
sudo pftcl -nf /etc/pf.conf
```

Vrlo važan parametar naredbi pfctl je `-s` nakon kojeg slijedi jedna od sljedećih ključnih riječi: `nat`, `queue`, `rules`, `Anchors`, `state`, `Sources`, `info`, `labels`, `timeouts`, `memory`, `Tables`, `osfp`, `Interfaces`, `all`. Za ilustraciju dosad postavljenih pravila primijenit će se nekoliko spomenutih vrsta informacija kako bi se dobili podaci o završenim vezama, trenutno uspostavljenim vezama (*keep state*), broju primljenih paketa i još nekim značajnijim karakteristikama.

```
FreeBSD# pfctl -s rules
block drop all
pass out proto tcp from any to any port = ssh keep state
pass out proto tcp from any to any port = smtp keep state
pass out proto tcp from any to any port = domain keep state
pass out proto tcp from any to any port = http keep state
pass out proto tcp from any to any port = pop3 keep state
pass out proto tcp from any to any port = auth keep state
pass out proto tcp from any to any port = pop3s keep state
pass out proto udp from any to any port = domain keep state
```

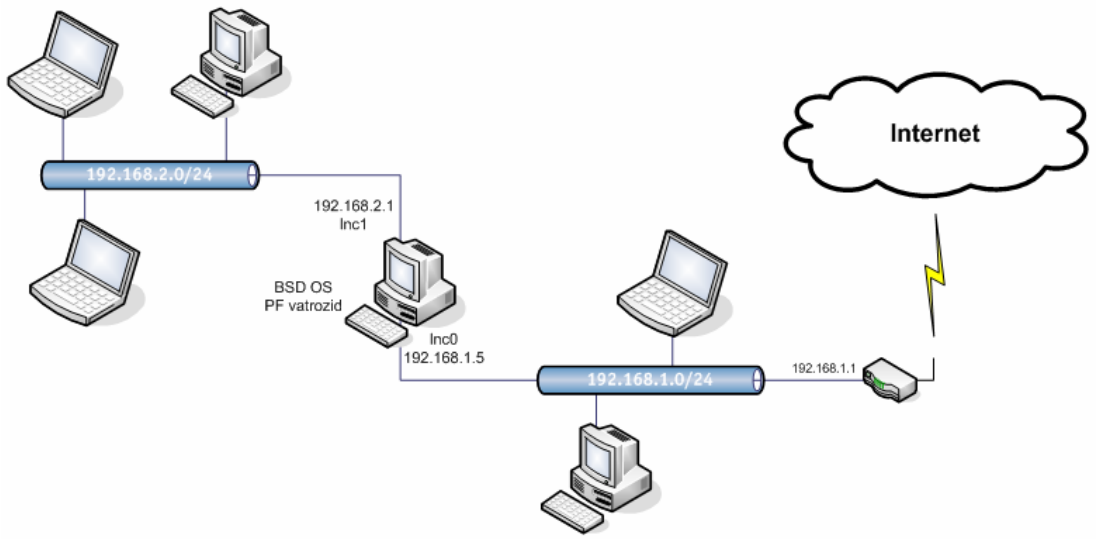
```
FreeBSD# pfctl -s state
self tcp 192.168.1.10:63044 -> 192.168.1.3:22    ESTABLISHED:ESTABLISHED
```

```
FreeBSD# pfctl -s info
Status: Enabled for 0 days 00:03:19           Debug: Urgent
Hostid: 0x7c8d54ca
State Table                          Total             Rate
  current entries                        2
  searches                            1064            5.3/s
  inserts                               67            0.3/s
  removals                              65            0.3/s
Counters
  match                                310            1.6/s
  bad-offset                             0            0.0/s
  fragment                               0            0.0/s
  short                                  0            0.0/s
  normalize                              0            0.0/s
  memory                                 0            0.0/s
  bad-timestamp                          0            0.0/s
  congestion                             0            0.0/s
  ip-option                              0            0.0/s
  proto-cksum                            0            0.0/s
  state-mismatch                         0            0.0/s
  state-insert                           0            0.0/s
  state-limit                            0            0.0/s
  src-limit                              0            0.0/s
  synproxy                               0            0.0/s
```

4.4. Konfiguracija poveznika

Poveznik (pristupnik, eng. *gateway*) je računalo ili usluga koja povezuje dvije ili više računalnih mreža koje se razlikuju po dodijeljenom adresnom segmentu, korištenim protokolima ili nekim drugim karakteristikama. Radi se o zastarjelom terminu koji je u početku označavao usmjerivače, a danas se koristi za imenovanje čitavog skupa usluga poput vatrozida, usmjerivača i sl.

Situacija se u ovom slučaju komplicira utoliko što računalo koje se konfigurira, a time i PF aplikacija, postaje jedina veza između dviju mreža i nosi odgovornost o prosljeđivanju paketa iz jedne na drugu mrežu (ili više). Za potrebe razumijevanja sljedećeg teksta potrebno je dogovoriti konvencije kako bi se olakšao zahtjevan posao podešavanja. Korištene su dvije mreže s adresama 192.168.1.0/24 i 192.168.2.0/24 i nekoliko računala. Pregledan opis zajedno sa svim potrebnim podacima dan je na sljedećoj skici.



Slika 1: Shematski prikaz mreže

Najprije je potrebno uključiti prosljeđivanje IP paketa. U naredbenoj liniji to je moguće učiniti naredbama:

```
sysctl net.inet.ip.forwarding=1
sysctl net.inet6.ip6.forwarding=1
```

Druga naredba odnosi se na novu inačicu adresiranja - IPv6 pa nije nužna, ali se zbog prelaska na nov način adresiranja preporuča pokretanje i te naredbe.
Kako se ne bi moralo nakon svakog pokretanja sustava činiti isto, moguće je upisati naredbe u odgovarajuće konfiguracijske datoteke. Za OpenBSD i NetBSD potrebno je u `/etc/sysctl.conf` datoteku upisati:

```
net.inet.ip.forwarding=1
net.inet6.ip6.forwarding=1
```

Na FreeBSD operacijskim sustavima unutar `/etc/rc.conf` datoteke treba upisati:

```
gateway_enable="YES"
ipv6_gateway_enable="YES"
```

Razlika je jedino u tome što FreeBSD nastoji sve postavke centralizirati unutar `/etc/rc.conf` datoteke.
Slijedi opis pravila koja određuju prosljeđivanje paketa s jednog sučelja na drugo, odnosno s jedne mreže na drugu. Najjednostavnije je preusmjeravanja s jedne mreže na sve ostale, postavljanjem sljedeće linije u `/etc/pf.conf`:

```
pass from lnc1:network to any port $port keep state
```

Pri tome se oznaka `$port` odnosi na listu portova čijim će se paketima dopuštati prolaz. Ukoliko se želi u potpunosti preusmjeravati promet s mreže na koju je računalo spojeno sučeljem `lnc1` na bilo koju drugu, treba izostaviti dio s portovima:

```
pass from lnc1:network to any keep state
```

U ovom će se slučaju, ukoliko se primjerice pokrene naredbu `ping` na jednom računalu iz svake mreže, uočiti propusnost u jednom smjeru. Za potpunu funkcionalnost očigledno je potrebno dodati pravilo koje omogućava prijenos paketa i u suprotnom smjeru:

```
pass from lnc0:network to any keep state
```

Takva dva pravila se mogu zamijeniti jednim:

```
pass from any to any keep state
```

što nije korisno u smislu sigurnosti, ali u svrhu ispitivanja temeljnih funkcionalnosti svakako je korisno.
Radi cjelovitosti valja spomenuti još dva pravila kojima se može odrediti slična funkcionalnost:

```
pass in on lnc1 from lnc1:network to lnc0:network port $ports keep state
pass out on lnc0 from lnc1:network to lnc0:network port $ports keep state
```

Kao što je vidljivo u prethodnim naredbama, PF vatrozid omogućuje imenovanje mreža korištenjem sintagme *sučelje:mreža* ili korištenjem IP adrese mrežnog segmenta uz mrežnu masku poput 192.168.1.0/24. Dodatno, korisnicima se savjetuje korištenje makro naredbi u svakoj prilici, a njihova potpuna funkcionalnost dolazi do izražaja kod složenijih konfiguracija.

4.5. Uključivanje NAT funkcionalnosti

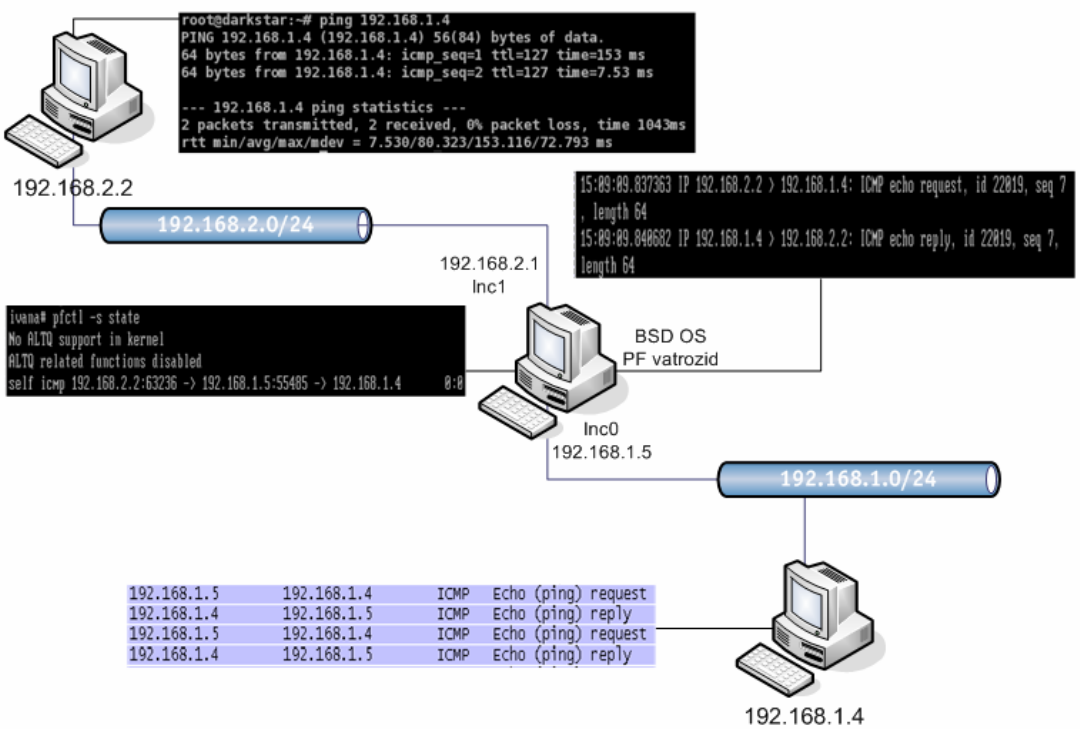
Nakon podešavanja poveznika, može se prijeći na uključivanje NAT funkcionalnosti. Najjednostavniji oblik je prosljeđivanje svih paketa uz izmjenu izvorne IP adrese. Da bi se to ostvarilo unutar konfiguracijske datoteke treba dodati retke:

```
nat on lnc0 from lnc1:network to any -> (lnc0)
pass all
```

Kako bi se ostalo u skladu s prethodno danim savjetima vezanim uz korištenje makro naredbi, treba prepisati ovaj kratki kôd u nešto duži, ali zato pregledniji:

```
ext_if="lnc0"
int_if="lnc1"
localnet=$int_if:network
nat on $ext_if from $localnet to any -> ($ext_if)
pass all
```

Na sljedećem prikazu dano je stanje na računalu iz lokalne mreže koje generira ICMP *echo* pakete naredbom `ping`, a s druge strane je ciljano računalo s mrežnim analizatorom koje prima *echo* zahtjeve i vraća ICMP *echoreply* odgovore.



Slika 2: Komunikacija korištenjem NAT mehanizma

Iz primjera se može vidjeti sintaksa naredbe `nat` koja je sama po sebi intuitivna osim što je potrebno naglasiti razlog pojavljivanja zagrada kod posljednjeg parametra. U ovom slučaju zagrade označavaju sučelje čija se IP adresa dinamički mijenja (DHCP) za razliku od slučajeva kad se IP adrese fiksno određuju. U tom slučaju pri promjeni IP adrese sučelja, PF ne bi prihvatio izmjenu i koristio bi staru adresu. Zato su određene zagrade kao eksplicitni pokazatelj mogućeg dinamičkog mijenjanja adrese.
Ukoliko je potrebno napraviti iznimku kod NAT pravila, nju je potrebno postaviti prije odgovarajućeg NAT pravila. Primjerice, u slučaju kada se prevode sve interne IP adrese iz segmenta 192.168.2.0/24 osim one s računala 192.168.2.100. Tada je potrebno napisati sljedeći niz pravila:

```
no nat on lnc0 from 192.168.2.100 to any
nat on lnc0 from 192.168.2.0/24 to any -> 161.53.64.222
```

Jedan od oblika prevođenja je i dvosmjerno prepisivanje adresa (eng. *bidirectional mapping*, *1:1 mapping*), a koristi se u slučajevima kada je potrebno posjedovati preslikavanje samo jedne IP adrese s lokalne mreže na jednu javnu, a ostale se, primjerice, preslikavaju na neku drugu. Vanjske veze na javnu adresu biti će pretvorene u internu, a jednako tako i unutarnje veze prema Internetu poput DNS upita, bit će pretvorene u veze s izvornom javnom adresom. Pravilo koje pretvara privatnu IP adresu 192.168.2.3 u javnu adresu 161.55.66.100 izgleda ovako:

```
binat on lnc0 from 192.168.2.3 to any -> 161.55.66.100
```

4.6. Preusmjeravanje

Još jedna od korisnih mogućnosti PF vatrozida je preusmjeravanje (redirekcija, eng. *redirection, port forwarding*). NAT funkcionalnost omogućuje korištenje zajedničke javne IP adrese nizu računala iz lokalne mreže. Međutim, kako s Interneta adresirati jedno od računala na lokalnoj mreži? Odgovor leži u korištenju mehanizma preusmjeravanja koji omogućuje prosljeđivanje dolaznog mrežnog prometa na određeno računalo unutar mreže. Takav oblik funkcionalnosti još se naziva i DNAT (eng. *Destination NAT*).
Za uvod se može napisati primjer koji preusmjerava sav nadolazeći promet usmjeren na port 80 na računalo s IP adresom 192.168.2.2. Na taj način je web poslužitelj (port 80) dostupan i vanjskom svijetu:

```
rdr on lnc0 proto tcp from any to any port 80 -> 192.168.2.2
```

Sljedeće naredbe prikazuju mogućnosti kod preslikavanja niza portova:

```
rdr on lnc0 proto tcp from any to any port 5000:5500 -> 192.168.2.2
rdr on lnc0 proto tcp from 27.155.166.0/24 to any port \
   5000:5500 -> 192.168.2.2 port 6000
rdr on lnc0 proto tcp from any to any port \
   5000:5500 -> 192.168.2.2 port 7000:*
```

Prvim pravilom preslikavaju se portovi od broja 5000 do 5500 i usmjeravaju na računalo s adresom 192.168.2.2. Drugo pravilo demonstrira vrlo korisnu mogućnost prosljeđivanja paketa na računalo s lokalne mreže i to samo ukoliko je izvorna adresa paketa iz mrežnog segmenta 27.155.166.0/24 te dodatno preslikava portove iz intervala 5000 do 5500 na port 6000. Posljednje pravilo slično je prvom osim što se port 5000 preslikava u 7000, 5001 u 7001, itd...
Svi oblici prevođenja IP adresa obavljaju se neposredno prije prolaska kroz filtar paketa pa o tome treba voditi računa. Naime, potrebno je pripaziti da se preusmjereni paketi dozvole i naredbama filtriranja prometa.
Postavljanje `rdr` pravila potencijalno dopušta pristup čitavoj lokalnoj mreži ukoliko računalo na koje se dolazeći promet usmjerava sadrži sigurnosni nedostatak. Taj problem se rješava stvaranjem demilitarizirane zone (eng. *DMZ - demilitarized zone, PSN - private service network*) unutar koje se smještaju sva računala koja trebaju biti vidljiva s Interneta ili neke druge nezaštićene mreže. Ukoliko je npr. web poslužitelj ranjiv, ostali dijelovi lokalne mreže nisu ugroženi jer se strogo prati promet između DMZ-a i ostatka mreže.
NAT mehanizam ima stanovite poteškoće u radu s FTP protokolom. Iako postoje sigurni protokoli za razmjenu podataka poput SFTP i SSH, ovaj važan protokol se ne smije i ne može izostaviti. Međutim, zbog karakteristika FTP veze, NAT ne može raditi na istom principu kao što funkcionira za ostale protokole i zato su ponuđena različita rješenja ovisna o operacijskom sustavu korisnika. Najprije je razvijen *FTP proxy* za PF. Slijedi razvoj *ftpsesame* i *pftpx* servisa te naposljetku *modern FTP-proxy* kao sastavni dio OpenBSD 3.9 i svih novijih distribucija.

4.7. Prepoznavanje operacijskih sustava

PF ima ugrađenu mogućnost prepoznavanja operacijskog sustava (eng. *Passive Operating System Fingerprinting*) koji je izvorište primljenog paketa, a za rad se koristi `/etc/pf.os` datoteka. Unutar konfiguracijske datoteke ključna riječ koja označava operacijski sustav je `os`. Naredba koja omogućava pregled pravila vezanih uz operacijske sustave glasi:

```
# pfctl -s osfp
```

Primjer pravila dan je na sljedećem ispisu.

```
pass  in on $ext_if from any os OpenBSD keep state
block in on $ext_if from any os "Windows 2000"
block in on $ext_if from any os "Linux 2.4 ts"
block in on $ext_if from any os unknown
```

Važno je napomenuti da ova funkcionalnost ne daje uvijek ispravne rezultate pa ju je potrebno koristiti s oprezom.

4.8. Filtriranje temeljeno na zastavicama

Ovakvo filtriranje najčešće se koristi kod TCP paketa koji se pojavljuju tijekom uspostavljanja veze. TCP zastavice i njihova značenja nalaze se na sljedećem popisu:

- F : FIN – *Finish*; kraj sjednice,
- S : SYN – *Synchronize*; označava zahtjev za uspostavom veze,
- R : RST – *Reset*; prekid uspostavljene veze,
- P : PUSH – *Push*; paket se šalje trenutno,
- A : ACK – *Acknowledgement*; potvrda primitka,
- U : URG – *Urgent*; oznaka hitnosti,
- E : ECE – *Explicit Congestion Notification Echo*; označava naprednu mogućnost kontrole zagušenja,
- W : CWR – *Congestion Window Reduced*; potvrda o primitku paketa s ECE zastavicom.

Pravilo ima sintaksu koja izgleda ovako:

```
flags check/mask
```

Prvi dio određuje koje zastavice moraju biti postavljene kako bi paket bio propušten, a drugi dio koje zastavice se provjeravaju. Sljedeće pravilo demonstrira korištenje spomenute funkcionalnosti:

```
pass in on fxp0 proto tcp from any to any port ssh flags S/SA
```

4.9. Obrana od paketa s lažnim izvornim IP adresama

Napadači često mijenjaju izvornu IP adresu paketima (eng. *IP spoofing*) kako bi skrili svoju ispravnu adresu ili kako bi se lažno predstavili. PF ima mehanizam za obranu od ovakvih napada, korištenjem ključne riječi `antispoof`.

```
antispoof for fxp0 inet
```

Ovo se pravilo prilikom učitavanja razdvaja na dva pravila, a pod pretpostavkom da je IP adresa 192.168.1.5/24 na `fxp0` sučelju, to izgleda ovako:

```
block in on ! fxp0 inet from 192.168.1.5/24 to any
block in inet from 10.0.0.1 to any
```

Prvo pravilo govori da se svi paketi s izvornom adresom iz 192.168.1.0/24 adresnog prostora trebaju odbaciti osim ako dolaze s `fxp0` sučelja. Drugo pravilo onemogućuje dolazak paketa s IP adrese

sučelja na isto sučelje jer računalo ne šalje samo sebi pakete korištenjem javne IP adrese. S ovim pravilom treba biti iznimno oprezan jer se nepravilnim korištenjem može zabraniti propuštanje prometa na svim sučeljima.

4.10. TCP *proxy* poslužitelj

PF ima mogućnost funkcioniranja u tzv. posrednom (eng. *proxy*) načinu rada. Radi se o tome da PF uspostavi TCP vezu na ispravan način u tri koraka (eng. *three way handshake*), a zatim nakon analize uspostavljene veze, novu vezu, identičnu prethodnoj, uspostavi s računalom kojeg štiti. Ukoliko se, primjerice, PF postavi kao vatrozid web poslužitelja, naredba u konfiguracijskoj datoteci koja uključuje ovu funkcionalnost može izgledati ovako:

```
pass in on $ext_if proto tcp from any to $web_server port www \
   flags S/SA synproxy state
```

Primjer pravila koja koriste isključivo mehanizme filtriranja paketa:

```
ext_if  = "fxp0"
int_if  = "dc0"
lan_net = "192.168.0.0/24"

# tablica koja sadrži sve IP adrese dane vatrozidu
table <firewall> const { self }

# nema filtriranja na povratnom (eng. loopback) sučelju
set skip on lo0

# normalizacija svih dolaznih paketa
scrub in all

# podrazumijevano se paketi odbacuju
block all

# zaštita od "spoofing" napada na internom sučelju
antispoof quick for $int_if inet

# dozvoliti jedino ssh vezu s lokalne mreže s pouzdanog računala,
# npr. 192.168.0.15
# paket s postavljenom TCP RST zastavicom se odašilje da bi zatvorio vezu
# s nepouzdanim računalima
# quick naredba omogućava prekid pretraživanja pravila ukoliko se pojavi
# paket koji zadovoljava ovo pravilo
block return in quick on $int_if proto tcp from ! 192.168.0.15 \
   to $int_if port ssh flags S/SA

# proslijediti sav promet sa i na lokalnu mrežu
pass in  on $int_if from $lan_net to any
pass out on $int_if from any to $lan_net

# proslijediti tcp, udp i icmp pakete na sučelje spojeno na Internet
# pratiti stanja na udp i icmp paketima te mijenjati na tcp ISN broj
# (eng. Initial Sequence Number)
pass out on $ext_if proto tcp all modulate state flags S/SA
pass out on $ext_if proto { udp, icmp } all keep state

# dozvoliti ssh veze uspostavljene s vanjskog sučelja ako nisu adresirane
# na vatrozid
# spremati zapise o inicijalnim paketima radi kasnije analize
# za uspostavu veze koristiti tcp syn proxy
pass in log on $ext_if proto tcp from any to ! <firewall> port ssh \
   flags S/SA synproxy state
```

5. Naprednija izrada pravila

U prethodnim odjeljcima pojašnjena je funkcionalnost vatrozida dovoljna u većini slučajeva. Međutim, PF vatrozid sadrži i mnoštvo drugih mogućnosti koje uvelike mogu pomoći u poboljšanju rada mreže i povećanju sigurnosti.

5.1. Podešavanje *run-time* postavki

Korištenjem ključne riječi `set` određuju se postavke vezane uz ponašanje vatrozida u radu. Te se postavke primjenjuju prilikom učitavanja pravila, a ukoliko se pojedinu izuzme iz `pf.conf` datoteke, koristi se njezina podrazumijevana vrijednost koja je u sljedećoj listi podvučena za svaku mogućnost.

- `set block-policy` ***option*** – određuje ponašanje kod neprihvaćanja paketa, odbacivanje (*drop*) ili vraćanje paketa s postavljenom zastavicom TCP RST (*return*),
- `set debug` ***option*** – različite razine prikaza poruka (*none*, *urgent*, *misc*, *loud*),
- `set fingerprints` ***file*** – datoteka s podacima o operacijskim sustavima, inicijalno `/etc/pf.os`,
- `set limit` ***option value*** – različita ograničenja (*frags* 5000, *src-nodes* 10000, *states* 10000),
- `set loginterface` ***interface*** – sučelje s kojeg se skupljaju statistički podaci (*none*),
- `set optimization` ***option*** – optimizacija vatrozida (*normal*, *high-latency*, *aggressive*, *conservative*),
- `set skip on` ***interface*** – određuje se sučelje na kojeg se ne odnose pravila PF vatrozida,
- `set state-policy` ***option*** – ponašanje ukoliko je uključena `keep state` mogućnost (*if-bound*, *group-bound*, *floating*),
- `set timeout` ***option value*** – (*interval* 10, *frag* 30, *src.track* 0).

Primjer za prethodno navedeno:

```
set timeout interval 10
set timeout frag 30
set limit { frags 5000, states 2500 }
set optimization high-latency
set block-policy return
set loginterface dc0
set fingerprints "/etc/pf.os.test"
set skip on lo0
set state-policy if-bound
```

5.2. Normalizacija paketa

Normalizacija paketa (eng. *packet normalization, scrubbing*) koristi se za izbjegavanje pojave nejednoznačnosti kod određivanja konačnog odredišta paketa. Ovaj postupak uključuje i spajanje fragmentiranih paketa u cjelinu kako bi se spriječili neki oblici napada te odbacivanje paketa s nepravilno postavljenim TCP zastavicama. Najjednostavniji oblik naredbe izgleda ovako:

```
Scrub in all
```

Ovo uključuje normalizaciju svih paketa na svim sučeljima. Razlozi zbog kojih se ne treba uvijek koristiti `scrub` mogućnost su između ostalih i to što neki sustavi očekuju primanje djelomično neispravnih paketa koji bi se u ovom slučaju odbacili. Radi cjelovitosti navedeno je nekoliko načina korištenja mehanizma normalizacije paketa:

```
scrub in on fxp0 all fragment reassemble min-ttl 15 max-mss 1400
scrub in on fxp0 all no-df
scrub on fxp0 all reassemble tcp
```

5.3. Logiranje prometa

Zapisivanje podataka o propuštenim i odbačenim paketima obavlja se korištenjem `pflogd` pozadinske aplikacije. Čitanjem s `pflog0` sučelja, navedena aplikacija zapisuje informacije o paketima u `/var/log/pflog` datoteku i to korištenjem `tcpdump` binarnog formata. Ključna riječ koja se koristi je `log`. Da bi se sačuvali podaci o proslijeđenim paketima, ključnu riječ treba dodati kod `NAT/rdr` pravila i pravila filtriranja. Treba primijetiti da se naredbu za logiranje može navesti jedino u sprezi s pravilima za propuštanje ili odbacivanje paketa. Ukoliko se u pravilu nalazi i `keep state` odredba, samo se prvi paket sprema ako se prethodno ne navede ključna riječ `all`. Primjer jedne takve naredbe izgleda ovako:

```
pass in log (all) on $ext_if inet proto tcp to $ext_if port 22 keep state
```

Za čitanje zapisanih podataka iz datoteke potrebno je koristiti `tcpdump` aplikaciju budući da se radi o binarnom zapisu nečitljivom korištenjem uobičajenih uređivača teksta. U nastavku je navedeno nekoliko primjera naredbi za čitanje prema određenim zahtjevima. Za pregled samo onih paketa koji odgovaraju određenom portu može se koristiti neka od sljedećih naredaba:

```
# tcpdump -n -e -ttt -r /var/log/pflog port 80
# tcpdump -n -e -ttt -r /var/log/pflog port 80 and host 192.168.1.3
# tcpdump -n -e -ttt -i pflog0 host 192.168.4.2
```

5.4. Postavljanje redova paketa

PF vatrozid posjeduje mehanizam koji omogućava postavljanje paketa u red i određivanje njihovih prioriteta (eng. *Packet Queueing and Prioritization*). Svi primljeni paketi ne mogu se obraditi u trenutku njihova primanja pa ih operacijski sustav postavlja u red. Potom sustav odlučuje koje pakete iz reda treba obraditi i kojim redoslijedom. Upravo ovo može u značajnijoj mjeri utjecati na performanse čitave računalne mreže. Mehanizam koji određuje koji će se red obrađivati i kojim redoslijedom naziva se raspoređivačem (eng. *scheduler*). U prethodnom tekstu već je spomenuta ALTQ programska implementacija mehanizma za rad s redovima. Taj paket postao je sastavnim dijelom PF vatrozida i stoga je za njegov rad potrebno pokrenuti i PF vatrozid. Implementacija ALTQ na OpenBSD sustavu pruža podršku za CBQ (eng. *Class Based Queueing*) i PRIQ (eng. *Priority Queueing*) raspoređivače. Također, postoji programska podrška za RED (eng. *Random Early Detection*) i ECN (eng. *Explicit Congestion Notification)*. Mehanizam postavljanja poruka u red uključuje se, primjerice, sljedećom linijom postavljenom u `pf.conf`:

```
altq on fxp0 cbq bandwidth 2Mb queue { std, ssh, ftp }
```

Naredba uključuje CBQ način postavljanja poruka u red na `fxp0` sučelju s propusnošću od 2Mb u sekundi i tri procesa-djeteta: STD, SSH i FTP.
Određivanje svojstava procesa djece obavlja se korištenjem ključne riječi `queue` te slijedi nakon `altq` pravila.

```
queue std bandwidth 50% cbq(default)
queue ssh bandwidth 25% { ssh_login, ssh_bulk }
      queue ssh_login bandwidth 25% priority 4 cbq(ecn)
      queue ssh_bulk bandwidth 75% cbq(ecn)
queue ftp bandwidth 500Kb priority 3 cbq(borrow red)
```

Postavlja se propusnost (eng. *bandwidth*) `std` reda na polovicu ukupne propusnosti što iznosi 1Mb u sekundi i isti je postavljen kao podrazumijevani (eng. *default*) red. Red naziva `ssh` podijeljen je u dva s nazivima `ssh_login` i `ssh_bulk`. Prvi red je većeg prioriteta, ali i manje propusnosti. Oba imaju uključenu mogućnost rane detekcije nagomilavanja. Posljednji red je `ftp` s propusnošću 500Kb u sekundi, prioritetom broj 3 te mogućnošću povećanja propusnosti ukoliko zatreba. Uključen je i mehanizam izbjegavanja nagomilavanja - RED.

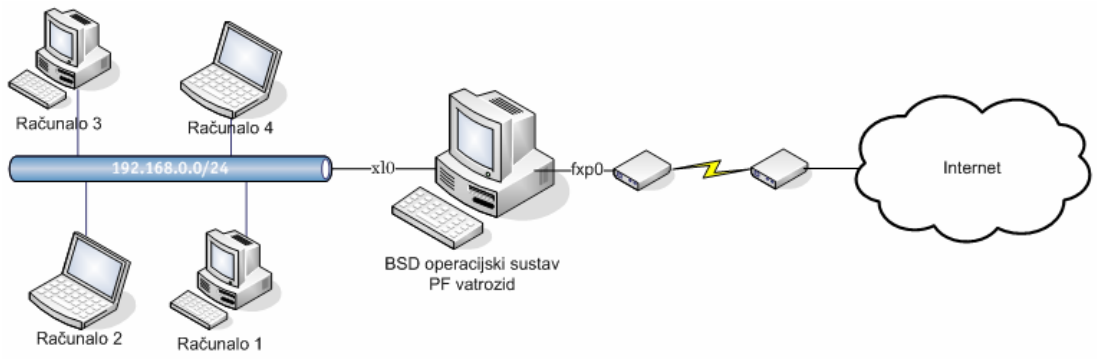
Konačno povezivanje nadolazećih poruka i reda obavlja se naredbom poput sljedeće:

```
pass out on fxp0 from any to any port 22 queue ssh
```

Korištenje mehanizma postavljanja paketa u red korisna je mogućnost koja zahtijeva pažljivo oblikovanje, ali zato donosi veliku potencijalnu korist u smislu ubrzanja i kvalitete rada računalne mreže.

6. Primjer oblikovanja pravila za manju računalnu mrežu

Kao primjer korištena je nešto drugačije oblikovana mreža nego u prethodnim primjerima. Lokalni adresni prostor je 192.168.0.0/24, a računalo s OpenBSD sustavom spojeno je `fxp0` sučeljem na Internet i `xl0` sučeljem na lokalnu mrežu.



Slika 3: Mreža koju PF vatrozid štiti

Prije poduzimanja bilo kakvih daljnjih koraka potrebno je odrediti koji je zapravo zadatak kojeg se želi dati PF vatrozidu:

- neometani pristup Internetu za svako računalo s lokalne mreže;
- podrazumijevano pravilo je "*deny*";
- dozvoli pristup vatrozidu s Interneta paketima sljedećih protokola:
  - SSH (TCP port 22),
  - *Auth/Ident*,
  - ICMP *Echo Request*;
- preusmjeriti sve zahtjeve s porta 80 vatrozida na računalo broj 3 i omogućiti uspostavu veze kroz vatrozid;
- zapisivati statističke podatke filtriranja s vanjskog sučelja;
- za sve odbačene veze odgovarati s TCP RST ili ICMP *Unreachable* paketima;
- niz pravila treba biti što jednostavniji u svrhu preglednosti i izmjena.

Podrazumijeva se prethodno pravilno postavljeno OpenBSD računalo u usmjerivačkom načinu rada (eng. *router mode*) te ispravan rad lokalne mreže i svih drugih računala koja su dio iste.
Na početku je potrebno oblikovati nekoliko makro naredbi radi jednostavnijeg korištenja:

```
ext_if="fxp0"
int_if="xl0"
tcp_services="{ 22, 113 }"
icmp_types="echoreq"
comp3="192.168.0.3"
```

Ukoliko se koristi PPPoE (eng. *Point-to-Point Protocol over Ethernet*) za spajanje na Internet, kao vanjsko sučelje se postavlja `tun0` umjesto `fxp0`.
Slijede pravila koja određuju način odbacivanja paketa i uključuju logiranje prometa:

```
set block-policy return
set loginterface $ext_if
```

Većina operacijskih sustava ima tzv. "*loopback*" povratno sučelje kojemu može pristupiti jedino računalo koje ga i posjeduje pa je razumno izuzeti to sučelje iz filtriranja:

```
set skip on lo
```

Preporuča se uključiti i mehanizam normalizacije paketa:

```
scrub in
```

Uključivanje NAT funkcionalnosti traži sljedeće pravilo:

```
nat on $ext_if from !($ext_if) to any -> ($ext_if)
```

Zagrade oko sučelja označavaju da je moguća dinamička promjena IP adrese sučelja, a izraz `!($ext_if)` označava sva sučelja osim `$ext_if` i ukoliko se doda još internih sučelja - neće biti potrebno mijenjanje pravila.
Da bi se omogućilo prometovanje FTP prometa korištenjem NAT funkcionalnosti, potrebno je uključiti niz pravila na način:

```
nat-anchor "ftp-proxy/*"
```

Preusmjeravanje je isto tako potrebno za FTP uslugu kako bi računala s lokalne mreže mogla uspostaviti veze s potrebnim računalima na Internetu. Pri tome se podrazumijeva korištenje isključivo porta 21:

```
rdr-anchor "ftp-proxy/*"
rdr on $int_if proto tcp from any to any port 21 -> 127.0.0.1 port 8021
```

Posljednje pravilo preusmjeravanja se koristi za preusmjeravanje zahtjeva pristiglih na port 80 na računalo broj 3 unutar lokalne mreže:

```
rdr on $ext_if proto tcp from any to any port 80 -> $comp3
```

Pravila filtriranja započinju pravilom za odbacivanje svih dolaznih paketa:

```
block in
```

Da bi sve veze uspostavljene od strane računala s lokalne mreže mogle ispravno funkcionirati potrebno je dodati pravilo:

```
pass out keep state
```

Pravila filtriranja uvode se na ovaj način:

```
anchor "ftp-proxy/*"
```

Zaštita od napada paketima s lažiranim adresama:

```
antispoof quick for { lo $int_if }
```

Potrebno je otvoriti sve portove važne za ispravan rad svih usluga:

```
pass in on $ext_if inet proto tcp from any to ($ext_if) port \
   $tcp_services flags S/SA keep state
```

Omogućiti prolaz kroz vatrozid paketima namijenjenih računalu broj 3:

```
pass in on $ext_if inet proto tcp from any to $comp3 port 80 \
   flags S/SA synproxy state
```

Može se dozvoliti prolaz i ICMP paketima:

```
pass in inet proto icmp all icmp-type $icmp_types keep state
```

Omogućiti još i prolaz iz lokalne mreže prema istoj:

```
pass in quick on $int_if
```

Konačno, slijedi potpun i funkcionalan niz pravila koja su upravo opisana:

```
# makro naredbe
ext_if="fxp0"
int_if="xl0"
tcp_services="{ 22, 113 }"
icmp_types="echoreq"
comp3="192.168.0.3"

# mogućnosti
set block-policy return
set loginterface $ext_if
set skip on lo

# scrub - normalizacija paketa
scrub in

# nat/rdr
nat on $ext_if from !($ext_if) -> ($ext_if:0)
nat-anchor "ftp-proxy/*"
rdr-anchor "ftp-proxy/*"
rdr pass on $int_if proto tcp to port ftp -> 127.0.0.1 port 8021
rdr on $ext_if proto tcp from any to any port 80 -> $comp3

# pravila za filtriranje
block in
pass out keep state
anchor "ftp-proxy/*"
antispoof quick for { lo $int_if }
pass in on $ext_if inet proto tcp from any to ($ext_if) port \
   $tcp_services flags S/SA keep state
pass in on $ext_if inet proto tcp from any to $comp3 port 80 \
   flags S/SA synproxy state
pass in inet proto icmp all icmp-type $icmp_types keep state
pass quick on $int_if
```

7. Zaključak

Korištenje vatrozida u svrhu zaštite pojedinih računala ili kompletnih mreža, danas više nije opcija nego nužnost. Kao dobar izbor zaštite nameće se odabir PF vatrozida. S obzirom na relativno jednostavan način podešavanja temeljnih funkcionalnost, PF je pogodan kako za zaštitu cijelih mreža, tako i za korištenje kod korisnika osobnih računala izravno spojenih na Internet. Naravno da se od korisnika pri tome očekuje određeno poznavanje BSD operacijskih sustava.

PF vatrozid posjeduje brojne funkcionalnosti, od kojih je dio opisan u ovom dokumentu. Od funkcionalnosti se ističe *stateful inspection* način filtriranja paketa koji je dijelom implementiran i za UDP protokol. Vrlo je korisna i mogućnost određivanja prioriteta za pojedine oblike mrežnog prometa korištenjem redova, kao i NAT funkcionalnost koja je podržana i u obliku redirekcije. Iako funkcionalnost prepoznavanja operacijskih sustava nije u potpunosti ispravna tj. vještiji korisnici je mogu zaobići, ta funkcionalnost sigurno u nekim pouzdanim mrežama može biti veoma korisna. Funkcionalnost logiranja je danas neizostavan dio svakog kompleksnijeg vatrozida pa je tako ta funkcionalnost ugrađena i u PF vatrozid.

U ovom dokumentu nije bilo moguće opisati sve funkcionalnosti PF vatrozida pa se stoga čitateljima preporuča pregledavanje stručne literature (dio naveden u referencama), dok se tehnički opisi mogu pronaći i u priručnicima koji se isporučuju zajedno s PF paketom.

8. Reference

[1.] CARNet CERT u suradnji s LS&S: Logiranje NAT prometa, CCERT-PUBDOC-2006-06-160, lipanj 2006.

[2.] Newbie's Guide to PF, http://www.thedeepsky.com/howto/newbie_pf_guide.php, prosinac 2006.

[3.] PF: The OpenBSD Packet Filter, http://www.openbsd.org/faq/pf/, prosinac 2006.

[4.] Peter N. M. Hansteen: Firewalling with OpenBSD's PF packet filter, http://www.bgnett.no/~peter/pf/, kolovoz 2006.

[5.] The OpenBSD Packet Filter (PF) and ALTQ, http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/firewalls-pf.html, prosinac 2006.

[6.] Jacek Artymiak: Building Firewalls with OpenBSD and PF, 2nd ed., 2005.

[7.] Priručnik za pfctl, http://www.openbsd.org/cgi-bin/man.cgi?query=pfctl&sektion=8, prosinac 2006.

[8.] OpenBSD firewall using pf, http://www.muine.org/~hoang/openpf.html, prosinac 2006.

[9.] Daniel Hartmeier: OpenBSD Packet Filter, http://www.benzedrine.cx/pf.html, prosinac 2006.