Filtrirajuci Mostovi

Alex Dupre

<ale@FreeBSD.org>

$FreeBSD: doc/en_US.ISO8859-1/articles/filtering-bridges/article.sgml,v 1.20 2004/08/08 13:43:54 hrs Exp $

Cesto je korisno da se jedna fizicka mreza (kao sto je Ethernet) podeli na dva odvojena segmenta bez potrebe za kreiranjem subnet maski, i koriscenjem rutera za njihovo spajanje. Uredjaj koji spaja dve mreze na ovaj nacin se naziva most (bridge). FreeBSD sistem sa dva mrezna interfejsa je dovoljan da bi mogao da radi kao most.
Most radi tako sto skenira adrese MAC nivoa (Ethernet adrese) uredjaja povezanih na svaki od njegovih mreznih interfejsa i onda prosledjuje saobracaj izmedju dveju mreza samo ako su izvor i odrediste na razlicitim segmentima. Sa vecine gledista most je slican Ethernet switch-u sa samo dva porta.

[Podeljeni HTML / Jedan HTML]

---

1 Zasto koristiti filtrirajuci most?

Sve vise i vise, zahvaljujuci snizenju cena broad band Internet konekcija (xDSL) i zbog smanjenja dostupnih IPv4 adresa, veliki broj kompanija je povezan sa Internetom 24 sata non-stop sa nekoliko (ponekad ne vise od 2) IP adresa. U ovakvim situacijama je cesto pozeljno imati zastitni zid koji filtrira dolazni i odlazni saobracaj, od, i prema Internetu, ali packet filtering resenje koje se zasniva na ruteru mozda nije odgovarajuce, ili zbog problema subnet-ovanja, ruter je vlasnistvo dobavljaca konekcije (ISP), ili zbog toga sto ne podrzava takve opcije. U ovakvim scenarijima se narocito preporucuje koriscenje filtrirajuceg mosta.
Zastitni zid zasnovan na Mostu moze biti konfigurisan u ubacen izmedju xDSL rutera i vaseg Ethernet hub/switch-a bez ikakvih problema IP numerisanja.

---

2 Kako instalirati

Dodavanje funkcije mosta FreeBSD sistemu nije tesko. Od 4.5 izdanja moguce je ucitati te funkcije kao module umesto potrebe da ponovo kompajlirate kernel, u mnogome pojednostavljujuci proceduru. U sledecim podsekcijama objasnicu oba instalaciona nacina.

Vazno: *Nemojte* pratiti obe instrukcije: jedna procedura *iskljucuje* onu drugu. Odaberite najbolji izbor prema vasim potrebama i mogucnostima.

Pre nego nastavimo, trbalo bi da imate najmanje dve Ethernet kartice koje podrzavaju promiscuous mod i za primanje i za prenosenje, zato sto moraju biti sposobni da salju Ethernet pakete sa bilo kojom adresom, ne samo sa njihovom. Plus da bi imali dobru propusnu moc, kartice trebaju biti PCI bus mastering kartice. Jos uvek su najbolji izbor Intel EtherExpress™ Pro, sa 3Com® 3c9xx serijom koja ih prati. Da bi ste pojednostavili konfiguraciju zastitnog zida moze biti korisno da imate dve kartice razlicitih proizvodjaca (koji koriste razlicite drajvere) da bi ste jasno odredili koji interfejs je povezan sa ruterom a koji sa unutrasnjom mrezom.

2.1 Konfiguracija Kernela

Znaci odlucili ste se da koristite stariji ali dobro iztestirani instalacioni metod. Da bi ste poceli, morate dodati sledece linije u konfiguracionom fajlu vaseg kernela:

```
options BRIDGE
options IPFIREWALL
options IPFIREWALL_VERBOSE
```

Prva linija kompajlira podrsku za most, druga je za zastitni zid i treca je za log funkcije zastitnog zida
Sada je potrebno izgraditi i instalirati novi kernel. Mozete naci detaljne instriukcije u sekciji Gradjenje i Instaliranje Custom Kernela FreeBSD Handbook-a.

2.2 Ucitavanje Modula

Ako ste odabrali da koristite novu i jednostavniju instalacionu metodu, jedina stvar koja treba da se uradi je da se doda sledeca linija u /boot/loader.conf:

```
bridge_load="YES"
```

Na ovaj nacin, prilikom podizanja sistema, bridge.ko modul ce se ucitati zajedno sa kernelom. Nije potrebno dodavati slicnu liniju za ipfw.ko modul, posto ce se on automatski ucitati nakon izvrsenja koraka u sledecoj sekciji.

---

3 Zavrsne Pripreme

Pre nego sto restartujete sistem da bi ucitali novi kernel ili trazene module (prema ranije odabranoj instalacionoj metodi), morate napraviti neke izmene u /etc/rc.conf konfiguracionom fajlu. Podrazumevano pravilo zastitnog zida je da blokira sve IP pakete. Za pocetak cemo podesiti otvoreni zastitni zid, da bi smo potvrdili njegovo delovanje bez nekih problema vezanih za filtriranje paketa (u slucaju da nameravate da izvrsite ovu proceduru udaljeno, takva konfiguracija ce izbeci da ostanete izolovani od mreze).
Stavite sledece linije u /etc/rc.conf:

```
firewall_enable="YES"
firewall_type="open"
firewall_quiet="YES"
firewall_logging="YES"
```

Prva linija ce omoguciti zastitni zid (i ucitace modul ipfw.ko ako nije kompajliran u kernelu), druga ga postavlja u otvoreni mod (kao sto je to objasnjeno u /etc/rc.firewall), treca linija nece prikazati pravila koja se ucitavaju i cetvrta linija da omoguci podrsku za logiranje.
Sto se tice konfiguracije mreznih interfessa, najcesce se koristi nacin dodeljivcanja IP samo jednoj mreznoj kartici, ali most ce raditi cak iako su oba interfejsa kofigurisala IP ili nijedna. U zadnjem slucaju (bez IP) most masina ce biti jos vise skrivenija, kojoj se nemoze pristupiti sa mreze: da bi je konfigurisali, morate se ulogovati iz konzole ili preko treceg mreznog interfejsa odvojenog od mosta. Nekada, prilikom podizanja sistema, neki programi zahtevaju mrezni pristup, na primer za domain resoluciju: u ovom slucaju je neophodno dodeliti IP spoljnom interfejsu (onom koji je povezan na Internet, gde su DNS serveri), zato sto ce se most aktivirati na kraju procedure podizanja sistema. To znaci da se fxp0 interfejs (u nasem slucaju) mora navesti u ifconfig sekciji /etc/rc.conf fajla, dok xl0 ne mora. Dodeljivanje IP za obe mrezne kartice nema mnogo smisla, osim ako, prilikom procedure podizanja sistema, aplikacije trebaju pristupiti servisima na oba Ethernet segmenta.
Postoji jos jedna vazna stvar koju treba da znate. Kada se koristi IP preko Etherneta, u stvari postoje dva Ethernet protokola koji se koriste: jedan je IP, a drugi je ARP. ARP radi konverziju IP adrese hosta u njegovu Ethernet adresu (MAC nivo). Da bi ste dozvolili komunikaciju dva hosta odvojenih mostom, neophodno je da most prosledjuje ARP pakete. Takav protokol nije ukljucen u IP nivo, posto postoji samo sa IP preko Etherneta. FreeBSD zastitni zid filtrira iskljucivo na IP nivou i stoga svi ne-IP paketi (gde spada i ARP) ce biti prosledjeni bez filtriranja, cak i u slucaju da je zastitni zid konfogurisan da nista ne dozvoli.
Sada je vreme da restartujete sistem i da ga koristite kao i ranije: popjavice se neke nove poruke vezane za most i zastitni zid, ali most nece biti aktiviran a zastitni zid, zato sto je u otvoreni modu, nece izbeci nijednu operaciju.

Ako postoje neki problemi, trebalo bi da ih resita sada pre nastavka.

---

4 Omogucavaje Mosta

Sada, da bi omogucili most, morate izvrsite sledece komande (zamenite imena mreznih interfejsa fxp0 i xl0 sa vasim):

```
# sysctl net.link.ether.bridge.config=fxp0:0,xl0:0
# sysctl net.link.ether.bridge.ipfw=1
# sysctl net.link.ether.bridge.enable=1
```

Prvi niz odredjuje koji ce interfejsi biti aktivirani od strane mosta, drugi ce omoguciti zastitni zid na mostu i konacno treci niz ce omoguciti most.

> **Beleska:** Ako imate FreeBSD 5.1-RELEASE ili raniji sysctl varijable su spelovane drugacije. Pogledajte most(4) za detalje.

Sada mozete ubacii vasu masinu izmedju dva seta hostova ne ugrozavajuci bilo kakvu komunikaciju izmedju njih. U tom slucaju, sledeci korak je dodavanje net.link.ether.bridge.*[blah]=[blah]* delova ovih nizova u /etc/sysctl.conf fajl, da bi se izvrsili pri startovanju.

---

5 Konfigurisanje Zastitnog Zida

Sada je vreme da kreirate vas fajl sa custom pravilima zastitnog zida, da bi ste osigurali unutrasnju mrezu. Postojace neke kompplikacije tokom rada na ovome zato sto nisu sve funkcije zastitnog zida dodstupne na bridged paketima. Postoji razlika izmedju paketa koji su u procesu prosledjivanja i paketa koje lokalna masina prima. Generalno, dolazni paketi prolaze kroz zastitni zid samo jednom, a ne dva puta kao sto je to normalno slucaj; u stvari oni se filtriraju samo nakon prijema, tako da pravila koja koriste out ili xmit se nikada nece podudariti. Licno, ja koristim in via koja je starija sintaksa, ali koja ima smisao kada je citate. Jos jedno ogranicenje je da ste ograniceni da koristite samo pass ili drop komande za pekete koje se filtriraju od strane mosta. Sofisticiranije stvari kao divert, forward ili reject nisu dostupne. Te opcije se i dalje mogu koristiti, ali samo za saobracaj do ili od samog mosta (ako ima IP adresu).

Novo u FreeBSD 4.0, je koncept stateful filtriranja. Ovo je veliki napredak za UDP saobracaj, koji je ustvari tipican zahtev koji ide van, pracen nakon toga odgovorom sa tim istim setom IP adresa i brojeva portova (ali sa obrnutim izvorom i odredistem naravno). Za zastitne zidove koji nemaju odrzanje veze, skoro da nepostoji nacin da obradjuju ovaj tip saobracaja kao pojedinacne sesije. Ali sa zastitnim zidom koji moze da “zapamti” izlazni UDP paket i, u nekoliko narednih minuta, dozvoli odgovor, rukovodjenje UDP servisima je trivijalno. Sledeci primer pokazuje kako se to radi. Moguce je uraditi istu stvar i sa TCP paketima. Ovo vam dozvoljava da izbegnete neke denial of service napade i ostale prljave trikove, ali isto tako tipicno cini da vasa stable tabela brzo naraste u velicini.

Pogledajmo primer podesavanja. Primetite prvo da na vrhu /etc/rc.firewall vec postoje standardna pravila za loopback interfejs lo0, tako da za to ne treba da brinemo. Custom pravila trebaju biti stavljena u poseban fajl (npr /etc/rc.firewall.local) i ucitana prilikom startovanja sistema, tako sto cete modifikovati niz u /etc/rc.conf gde smo definisali otvoreni zastitni zid:

```
firewall_type="/etc/rc.firewall.local"
```

> **Vazno:** Trebate navesti *punu* putanju, u suprotnom se nece ucitati, sa rizikom da ostane izolovan od mreze.

Zbog primera zamislite da imate fxp0 interfejs povezan prema spolja (Internet) i xl0 prema unutra (LAN). Most masina ima IP 1.2.3.4 (nemoguce je da vam vas ISP da klasu adresa A kao ovde, ali je dobro za nas primer).

```
# Things that we have kept state on before get to go through in a hurry
add check-state

# Throw away RFC 1918 networks
add drop all from 10.0.0.0/8 to any in via fxp0
add drop all from 172.16.0.0/12 to any in via fxp0
add drop all from 192.168.0.0/16 to any in via fxp0
```

```
# Allow the bridge machine to say anything it wants
# (if the machine is IP-less do not include these rows)
add pass tcp from 1.2.3.4 to any setup keep-state
add pass udp from 1.2.3.4 to any keep-state
add pass ip from 1.2.3.4 to any

# Allow the inside hosts to say anything they want
add pass tcp from any to any in via xl0 setup keep-state
add pass udp from any to any in via xl0 keep-state
add pass ip from any to any in via xl0

# TCP section
# Allow SSH
add pass tcp from any to any 22 in via fxp0 setup keep-state
# Allow SMTP only towards the mail server
add pass tcp from any to relay 25 in via fxp0 setup keep-state
# Allow zone transfers only by the slave name server [dns2.nic.it]
add pass tcp from 193.205.245.8 to ns 53 in via fxp0 setup keep-state
# Pass ident probes.  It is better than waiting for them to timeout
add pass tcp from any to any 113 in via fxp0 setup keep-state
# Pass the "quarantine" range
add pass tcp from any to any 49152-65535 in via fxp0 setup keep-state

# UDP section
# Allow DNS only towards the name server
add pass udp from any to ns 53 in via fxp0 keep-state
# Pass the "quarantine" range
add pass udp from any to any 49152-65535 in via fxp0 keep-state

# ICMP section
# Pass 'ping'
add pass icmp from any to any icmptypes 8 keep-state
# Pass error messages generated by 'traceroute'
add pass icmp from any to any icmptypes 3
add pass icmp from any to any icmptypes 11

# Everything else is suspect
add drop log all from any to any
```

Oni od vas koji imaju iskustva sa podesavanjem zastitnih zidova ce primetiti da neke stvari nedostaju. Tacnije, ne postoje anti-spoofing pravila, u stvari *nismo* dodali:

```
add deny all from 1.2.3.4/8 to any in via fxp0
```

Sto znaci, ispusti pakete koji dolaze od spolja a koji tvrde da dolaze sa nase mreze. Ovo je nesto sto ce te cesto uraditi da bi bili sigurni da neko ne pokusa da izbegne paket filter, generisuci zlocinacke pakete koji izgledaju kao da dolaze od iznutra. Problem sa ovim je da posoji *barem* jedan host na izlaznom interfejsu koji ne zelite da ignorisete: ruter. Ali obicno, ISP ima anti-spoof na njihovom ruteru, tako da nemoramo da se mnogo opterecujemo.

Zadnje pravilo izgleda kao pravi duplikat default pravila, sto znaci, ne dozvoli da ista prodje a da nije posebno dozvoljeno. Ali postoji razlika: sav sumnjivi saobracaj ce biti belezen.

Postoje dva pravila za propustanje SMTP i DNS saobracaja prema mail serveru i serveru imena, ako ih imate. Zapravo ceo skup pravila treba biti promenjen prema licnom ukusu, ovo je samo specifican primer (format pravila je tacno opisan u ipfw(8) man stranici). Primetite da da bi “relay” i “ns” radili, lookup servera imena mora da rade *pre* nego je most omogucen. Ovo je primer kako mozete videti da li ste podesili IP na tacnoj mreznoj kartici. Alternativno je moguce da odredite IP adresu umesto host imena (potrebno ako masina nema IP).

Oni koji su navikli da podesavaji zastitne zidove su navikli i da imaju pravilo za reset ili prosledjivanje ident paketa (TCP port 113). Nazalost, ova opcija se ne moze primeniti sa mostom, tako da bi najbolja stvar jednostavno bila da ih propustite do njihovog

odredista. Dokle kod ta odredisna masina nema tekuci ident daemon, ovo je relativno bezopasno. Alternativa je obaranje konekcije na portu 113, sto kreira neke probleme sa servisima kao npr IRC (ident probe mora da se ugasi).

Jedina druga stvar koja je malo cudna koju ste mozda primetili je da postoji pravilo koje dozvoljava masini sa mostom da komunicira, i jos jedno za unitrasnji host. Setite se da je ovo zbog toga sto ce dva seta saobracaja imati razlicite putanje kroz kernel i i u paket filter. Unutrasnja mreza ce ici kroz most, dok ce lokalna masina koristiti normalni IP stack za komunikaciju. Plus dva pravila za upravljanje razlicitim slicajevima. in via fxp0 pravila rade za obe putanje. Generalno, ako koristite in via pravila kroz filter, moracete da napravite izuzetak za lokalno generisane pakete, zato sto nisu dosli preko nijednog od vasih interfejsa.

---

6 Saradnici

Vecina delova ovog clanka je preuzeta, obnovljena i adaptirana iz starijeg teksta o mostovima, napisanog od strane Nick Sayer-a. Nekoliko inspiracija je nastalo nakon predstavljanja mostova od strane Steve Peterson-a.

Veliko hvala Luigi Rizzo-u za implementaciju most koda u FreeBSD i za vreme koje mi je posvetio odgovarajuci na sva moja pitanja.

Jos jedfno hvala ide Tom Rhodes-u koji je nadgledao moje prevodjenje sa Italijanskog (originalni jezik ovog clanka) na Engleski jezik.

~Dalibor Gudzic @2006