Dialup-firewalling sa FreeBSD-om

Marc Silver

marcs@draenor.org

$Date: 2002-02-06 17:01:34 $

Ovaj članak dokumentuje kako podesiti firewall koristeći PPP dialup sa FreeBSD-om i IPFW-om, kao i posebno podešavanje firewall-a na dialup sa dinamički dodeljenim IP adresama. Ovaj dokument ne pokriva podešavanje same PPP konekcije.

1 Uvod

Dialup-firewalling sa FreeBSD-om

Ovaj dokument pokušava da objasni postupak koji je potreban da bi se podesio firewall na FreeBSD-u u slučaju kada imate dinamički dodeljenu IP adresu. Mnogo truda je bilo uloženo u to da ovaj dokument budeš što informativniji i tačniji, svaki vaš komentar/sugestija autoru (mailto:marcs@draenor.org) je dobrodošla.

2 Opcije pri konfigurisanju kernela

Prva stvar koju morate da uradite jeste da rekompajlirate kernel FreeBSD-a. Ako vam je potrebno više informacija o tome kako da rekompajlirate kernel, onda je svakako najbolje mesto da počnete poglavlje o konfigurisanju kernela u Priričniku (http://www.freebsd.org/handbook/kernelconfig.html). Morate iskompajlirati kernel sa sledećim opcijama:

`options IPFIREWALL`

Ukljucuje firewall kod u kernel.

`options IPFIREWALL_VERBOSE`

Šalje logovane pakete system logger-u.

`options IPFIREWALL_VERBOSE_LIMIT=`*`100`*

Limitira broj ekvivalentnih unosa u log. Ovo sprečava da se vaš log prepuni sa mnogo jednakih unosa. *`100`* je razuman broj, ali možete promeniti ovaj podatak prema vašim potrebama.

`options IPDIVERT`

Omogućava *preusmeravanje* socket-a, o čemu će kasnije biti reči.

Postoje jos neke opcije koje nisu obavezne, a koje možete ukompajlirati u vaš kernel kako bi ste poboljšali bezbednost vašeg sistema. Ove opcije nisu neophodne, ali će neki paranoidniji korisnici svakako želeti da ih upotrebe.

`options TCP_RESTRICT_RST`

Ova opcija blokira sve TCP RST pakete. Ovo je najbolje koristiti za sisteme koji bi mogli biti izloženi SYN flooding-u (IRC serveri su dobar primer), ili sisteme za koje je poželjno da im se ne mogu lako skenirati portovi.

`options TCP_DROP_SYNFIN`

Ako uključite ovu opciju u kernel, biće ignorisani sci SYN i FIN paketi. Ovo onemogućava alate kao sto je nmap, itd. da odrede TCP/IP stek masine, ali onemogućava podršku za RFC1644 ekstenzije. Ovo NIJE preporučljivo ako će mašina raditi kao web server.

Nemojte odmah da restartujete mašinu pošto rekompajlirate kernel. Uz malo sreće moraćete samo jednom da uradite restartovanje da bi ste zavrsili instalaciju firewall-a.

3 Izmene `/etc/rc.conf` fajla potrebne za podizanje firewall-a

Sada je potrebno da uradimo odredjene izmene na `/etc/rc.conf` fajlu kako bi smo omogućili podizanje firewall-a. Jednostavno dodajte sledeće linije:

```
firewall_enable="YES"
firewall_script="/etc/firewall/fwrules"
natd_enable="YES"
natd_interface="tun0"
natd_flags="-dynamic"
```

Za više informacija o tome pogledajte u `/etc/defaults/rc.conf` i procitajte rc.conf(5)

4 Iskljcivanje PPP prevodjenja adresa

Moguće je da već koristite prevodjenje adresa (network address translation-NAT) koji je ugrađen u PPP. U tom slučaju moraćete da ga isključite, kao sto sledeći primeri koriste natd(8) da bi uradili istu stvar.

Ako već imate podešen ppp, verovatno imate blok instrukcija za automatsko podizanje PPP-a i to verovatno izgleda otprilike ovako:

```
ppp_enable="YES"
ppp_mode="auto"
ppp_nat="YES"
ppp_profile="profile"
```

Ako je tako, obrišite `ppp_nat="YES"` line. Takodje ćete morati da uklonite `nat enable yes` ili `alias enable yes` u `/etc/ppp/ppp.conf` fajlu.

5 Podešavanje firewall-a

Sada smo već pri kraju. Sve što je preostalo jeste da definišemo pravila po kojim će se firewall ponašati i posle toga možete da restartujete mašinu i firewall bi trebao da bude podignut i aktivan. Svakako razumem da će svako želeti nešto drugačija pravila za njihov firewall, tako da sam pokušao da napišem pravila koja će odgovarati većini korisnika dialup konekcije. Možete lako modifikovati već postojeća pravila kao osnovu za vaš set pravila. Krenimo prvo sa osnovama zatvorenog firewall-inga. Ono što bi trebalo da uradite jeste da u osnovnoj konfiguraciji sve zatvorite i onda po potrebi otvorite samo ono što vam je potrebno. Pravila se pišu tako da se prvo pišu pravila dozvole (allow), a onda pravila zabrane (deny). Pretpostavka je da ćete prvo dodati pravila dozvole, a zatim sve ostalo zabraniti. :)

Napravimo sada direktorijum /etc/firewall. Udjite u taj direktorijum i potom napravite file fwrules koji smo naznačili u rc.conf fajlu i editujte ga prema vašim potrebama. Molim vas da obratite pažnju na to da možete da promenite ime ovog fajla kako vama odgovara. Ovo uputsvo samo daje ovo ime kao primer.

Pogledajmo sada primer konfiguracionog fajla, gde ćete videti i detalje podešavanja:.

```
# Pravila za konfigurisanje firewall-a
# Written by Marc Silver (marcs@draenor.org)
# http://draenor.org/ipfw
# Slobodna distribucija


# Definišite firewall komadu (kao u /etc/rc.firewall) radi
# lakšeg referenciranja. Omogućava lakše čitanje istog .
fwcmd="/sbin/ipfw"

# Prisiljava preuzimanje novih pravila prilikom restartovanja.
$fwcmd -f flush

# Preusmerava sve pakete kroz tunnel interfejs.
$fwcmd add divert natd all from any to any via tun0

# Dozvoljava protok svih podataka kroz mrežne karte i localhost.
# Proverite da li ste promenili identifikaciju mrežne karte pre
# nego što restartujete mašinu (moja je bila fxp0) :)
$fwcmd add allow ip from any to any via lo0
$fwcmd add allow ip from any to any via fxp0

# Dozvoli sve konekcije koje ja iniciram
$fwcmd add allow tcp from any to any out xmit tun0 setup

# Pošto su konekcije napravljene, dozvoli im da ostanu otvorene
$fwcmd add allow tcp from any to any via tun0 established

# Svima na internetu je dozvoljeno da se konektuju na sledeće
# servise na ovoj mašini. U ovom primeru ljudi mogu da se
# konektuju na ssh i apache (ili neki drugi web server koji
# sluša na portu 80)
$fwcmd add allow tcp from any to any 80 setup
$fwcmd add allow tcp from any to any 22 setup

# Ova linija šalje RESET svim ident paketima.
$fwcmd add reset log tcp from any to any 113 in recv tun0
```

```
# Omogući izlazne DNS upite SAMO ka određenim serverima.
$fwcmd add allow udp from any to x.x.x.x 53 out xmit tun0

# Dozvoli im da daju povratnu infomaciju.... :)
$fwcmd add allow udp from x.x.x.x 53 to any in recv tun0

# Dozvoli ICMP (potreban za ping i traceroute). Možda ćete
# želeti da ovo onemogućite, ali to mojim potrebama odgovara
$fwcmd add 65435 allow icmp from any to any

# Zabrani sve ostalo.
$fwcmd add 65435 deny log ip from any to any
```

Sada imate potpuno funkcionalan firewall koji ce dozvoliti konekcije na portove 80 i 22, i koji će prijaviti pokušaj konekcije na bilo koji drugi port. Sada bi trebalo da možete bezbedno da restartujete mašinu i posle restarta bi vaš firewall trebao fino da radi. Ako pronadjete neku grešku ili natrčite na neki problem, ili imate bilo koju sugestiju kako bi unapredio ovu dokumentaciju, molim vas da mi napišete email.

6 Pitanja

1. Zašto koristite natd i ipfw kada bi mogli da koristite ugrađene ppp filtre?

Moraću da budem iskren na ovom mestu i da kažem da nemam određen razlog zašto zaista koristim natd i ipfw umesto ugradjenih ppp filtera. Posle razgovora koje sam imao sa raznim ljudima, došli smo do konsezusa da je ipwf definitivno moćniji kao i konfigurabilniji od ppp filtera sto mu daje poen više za funkcionalnost, kao i poen manje za lakoću korišćenja. Jedan od razloga zbog kojeg više volim ipwf jeste što preferiram da se firewalling radi na nivou kernela a ne na nivou korisničkog programa.

2. Ako interno koristim privatnu adresu, kao npr. u 192.168.0.0 opsegu, mogu li dodati komandu kao npr. $fwcmd add deny all from any to 192.168.0.0:255.255.0.0 via tun0 u pravila kako bih sprečio pokušaj konekcija sa udaljene mašine na internu mašinu?

Jednostavan odgovor je ne. Razlog je to sto natd radi prevodjenje adresa za *sve* sto je preusmereno kroz tun0. Sto se natd-a tiče, dolazeći paketi će govoriti samo o dinamički dodeljenoj IP adresi a NE o internoj mreži. Primetimo ipak da firewall-u možete dodati pravilo nalik na `$fwcmd add deny all from any to 192.168.0.0:255.255.0.0 via tun0` koje bi ograničilo host u vašoj internoj mreži da izađe uz pomoć firewall-a.

3. Mora da je nešto pogrešno. Pratio sam vaša uputstva od reči do reči i sada sam potpuno zatvoren.

Ovaj tutorial pretpostavlja da koristite *userland-ppp*, i iz tog razloga dati set pravila radi na `tun0` interfejsu, to odgovara prvoj konekciji ostvarenoj sa ppp(8) (odn. *user-ppp*). Dodatne konekcije bi koristile `tun1`, `tun2` itd.

Takođe bi trebalo da primetite da pppd(8) koristi ppp0 `ppp0` interfejs, tako da ako zelite da ostvarite konekciju pomocu pppd(8)-a morate zameniti `tun0` sa `ppp0`. Brz način da podesite firewall u tom slučaju je prikazan dole. Originalni set pravila je sačuvan `fwrules_tun0`.

```
% cd /etc/firewall
/etc/firewall% su
Password:
```

```
/etc/firewall# mv fwrules fwrules_tun0
/etc/firewall# cat fwrules_tun0 | sed s/tun0/ppp0/g > fwrules
```

Da bi ste saznali da li trenutno koristite ppp(8) ili pppd(8) mozete pogledati izlaz ifconfig(8)-a kada uspostavite vezu. Pod uslovom da uspostavite konekciju sa pppd(8)-om videćete nešto kao (prikazane su samo relevatne linije):

```
% ifconfig
(skipped...)
ppp0: flags=8051<UP,POINTOPOINT,RUNNING,MULTICAST> mtu 1524
        inet xxx.xxx.xxx.xxx --> xxx.xxx.xxx.xxx netmask 0xff000000
(preskočeno...)
```

OPod uslovom da je konekcija uspostavljena sa ppp(8)-om (*user-ppp*) trebalo bi da dobijete nesto nalik na slede'e linije:

```
% ifconfig
(skipped...)
ppp0: flags=8010<POINTOPOINT,MULTICAST> mtu 1500
(skipped...)
tun0: flags=8051<UP,POINTOPOINT,RUNNING,MULTICAST> mtu 1524
        (IPv6 stuff skipped...)
        inet xxx.xxx.xxx.xxx --> xxx.xxx.xxx.xxx netmask 0xffffff00
        Opened by PID xxxxx
(preskočeno...)
```