

# Tűzfal létrehozása betárcsázós kapcsolatokhoz FreeBSD-vel

Marc Silver

marcs@draenor.org

\$FreeBSD: doc/hu\_HU.ISO8859-2/articles/dialup-firewall/article.sgml,v 1.4  
2008/09/11 11:55:06 pgj Exp \$

A FreeBSD a FreeBSD Foundation bejegyzett védjegye.

A gyártók és terjesztők által használt megnevezések közül sok védjegy jogot követel. Ahol ilyen megnevezés tűnik fel ebben a dokumentumban, és a FreeBSD Projektnek tudomása volt a védjegyről, a megnevezést a "TM" vagy a "®" szimbólum követi.

Ebben a cikkben bemutatjuk, hogyan lehet beállítani tűzfalat a PPP-típusú kapcsolatokhoz a FreeBSD valamint az IPFW segítségével, különös tekintettel az olyan esetekre, ahol dinamikusan kiosztott IP-címmel használjuk a rendszert. Ez a leírás azonban nem tartalmazza magának a PPP-kapcsolatnak a beállítását. A PPP-kapcsolatok létrehozásához kérjük tekintse át a ppp(8) man oldalt.

Fordította: Páli Gábor <pgj@FreeBSD.org>

## 1. Bevezetés

A leírásban felvázoljuk azokat a lépéseket, amelyek szükségesek az Internet szolgáltatónk által dinamikusan kiosztott IP címmel rendelkező rendszerünk tűzfalának kiépítéséhez. Habár ezen cikk szerzője minden megtett, hogy ez a leírás minél hasznosabb és pontosabb legyen, örömmel várja az esetleges megjegyzéseket és javaslatokat a <marcs@draenor.org> címen.

## 2. Beállítások a rendszermagban

Az IPFW használatához bele kell fordítani némi támogatást a rendszer magjába. Ha többet szeretne tudni a rendszermag újrafordításáról, kérjük, olvassa el a a rendszermag beállításáról szóló fejezetet a Kézikönyvben ([http://www.FreeBSD.org/doc/hu\\_HU.ISO8859-2/books/handbook/kernelconfig.html](http://www.FreeBSD.org/doc/hu_HU.ISO8859-2/books/handbook/kernelconfig.html)). Az IPFW támogatásához az alábbi sorokat kell még hozzáírni a rendszermag konfigurációs állományához:

```
options IPFIREWALL
```

Elérhetővé teszi a rendszermag tűzfalért felelős rutinjait.

**Megjegyzés:** A cikk a FreeBSD 5.X-es verziójának használatát feltételezi. Azoknak a felhasználóknak, akik még a FreeBSD 4.X-es verzióját használják, a rendszermagjukat a *IPFW2* támogatással kell újrafordítaniuk. A FreeBSD 4.X felhasználóknak továbbá javasolt elolvasniuk ezzel kapcsolatosan a *ipfw(8)* man oldalt, kiemelten odafigyelve a *IPFW2 HASZNÁLATA A FreeBSD-STABLE-ben* fejezetre.

```
options IPFIREWALL_VERBOSE
```

Naplózott csomagok küldése a rendszernaplóba.

```
options IPFIREWALL_VERBOSE_LIMIT=500
```

Korlátozza az egyező tartalmú sorok naplózásának mennyiségét. Ezzel lehetővé válik, hogy a rendszernapló elárasztásának kockázata nélkül naplózzuk a tűzfal minden egyes tevékenységét, például egy "denial of service" (DoS) típusú támadás esetén. Itt az *500* egy viszonylag jó kiindulási érték lehet, de nyugodtan változtathajuk igényeink szerint.

**Figyelem** Amikor a rendszermag újrafordítása befejeződött, *ne indítsuk újra egyből* a rendszerünket. Ha így cselekszünk, könnyedén kizárhatjuk magunkat belőle! Csak azután szabad újraindítanunk és ezzel működésbe hozni a tűzfalat, miután a hozzátartozó szabályok a megfelelő helyre kerültek és minden hozzájuk kapcsolódó konfigurációs állományt megfelelően beállítottunk.

### 3. Az `/etc/rc.conf` módosítása a tűzfal betöltéséhez

Az `/etc/rc/rc.conf` konfigurációs állományt kell némileg átírnunk a tűzfal betöltéséhez, valamint a hozzátartozó szabályokat tartalmazó állomány helyének megadásához. Adjuk tehát hozzá az alábbi sorokat a `/etc/rc/rc.conf`-hoz:

```
firewall_enable="YES"
firewall_script="/etc/firewall/fwrules"
```

Ha többet szeretne tudni ezeknek a soroknak a jelentéséről, akkor nézze át a `/etc/defaults/rc.conf` állományt és olvassa el a `rc.conf(5)` man oldalt.

### 4. A PPP-ben levő címfordítás bekapcsolása

Amennyiben a helyi hálózatunkban fellelhető további kliensek számára is szeretnénk elérhetővé tenni az Internetet az átjárónkon át, szükségünk lesz a PPP-ben található hálózati címfordítás (Network Address Translation, NAT) beindítására. Ezt az `/etc/rc.conf`-ben a következő sorok hozzáadásával tehetjük meg:

```
ppp_enable="YES"
```

```
ppp_mode="auto"  
ppp_nat="YES"  
ppp_profile="internet_beallitasok"
```

**Megjegyzés:** Ne felejtjük el kicserélni az `internet_beallitasok` értékét a saját betárcsázós beállításait tartalmazó állomány nevére! Ennek nevének meg kell egyeznie a beállításaink `/etc/ppp/ppp.conf` állományban szereplő nevével.

## 5. A tűzfal szabályai

Most fogjuk megadni a rendszerünk tűzfalának szabályait. Az itt ismertetésre kerülő szabályok egy olyan általános sablont kívánnak bemutatni, amely a legtöbb betárcsázós felhasználó számára megfelelnek. Habár kétségtelen, hogy nem fogja mindenki igényeit tökéletesen kielégíteni, azonban segít megmutatni az IPFW működésének alapelveit és könnyedén tovább is fejleszthető.

Elsőként kezdjük a "zárt tűzfal" alapjaival. A zárt tűzfal lényegében azon a feltevésen alapszik, hogy alapvetően mindent kizárunk a rendszerből. Ezt követően a rendszergazda egyesével megadhatja azokat szabályokat, amelyeket engedélyezni kíván valamit. A szabályok közül először mindig azokat adjuk meg, amikkel engedélyezünk, majd azokat, amikkel tiltunk. Az alapfeltételezés szerint tehát a szabályokkal megadunk mindent, amit engedélyezünk a tűzfalon, és minden más pedig automatikusan tiltásra kerül.

Ezt követően hozzunk létre egy könyvtárat, ahol majd tárolni a fogjuk a tűzfalunk beállításait. Ebben a példában a `/etc/firewall/` könyvtárat fogjuk használni erre a célra. Lépjünk be ebbe a könyvtárba és hozzunk létre egy `fwrules` nevű állományt, ahogy azt az `rc.conf`-ban is megadtuk. Természetesen ez az elnevezés sem kötött, nyugodtan megváltoztathatjuk bármire. A leírás pusztán csak egy példát ad erre.

Most pedig nézzünk egy megjegyzésekkel tűzdelt szabályokat tartalmazó állományt:

```
# Definiálunk egy parancsot a tűzfalat összeállító program elérésére  
# (ld. /etc/rc.firewall). Remélhetőleg így könnyebb is lesz olvasni.  
fwcmd="/sbin/ipfw"  
  
# Megadjuk a külső hálózati csatolót. Ha felhasználói ppp-t használunk,  
# akkor ez valószínűleg a tun0 lesz.  
oif="tun0"  
  
# Megadjuk a belső hálózati csatolót. Ez többnyire (a helyi hálózaton  
# is elérhető) hálózati kártyánk lesz. Mindenképpen ellenőrizzük, hogy  
# jóladtuk-e meg!  
iif="fxp0"  
  
# Töröltsünk a rendszerben jelenleg érvényben levő össze szabályt,  
# még mielőtt betöltenénk a sajátjainkat.  
$fwcmd -f flush  
  
# Ellenőrizzük az összes csomag állapotát.  
$fwcmdl add check-state  
  
# Tiltsuk le az elrejtést a külső csatolón.
```

```
$fwcmd add deny ip from any to any in via $oif not verrevpath

# Engedélyezzünk minden általunk kezdeményezett kapcsolatot és
# tartsuk is meg az állapotukat. Ellenben tiltsunk minden olyat,
# amihez nincs semmilyen dinamikus szabály.
$fwcmd add allow ip from me to any out via $oif keep-state
$fwcmd add deny tcp from any to any established in via $oif

# Engedélyezzünk minden kapcsolatot a helyi hálózaton.
$fwcmd add allow ip from any to any via $iif

# Engedélyezzük a helyi (gépen belüli) forgalmat.
$fwcmd add allow all from any to any via lo0
$fwcmd add deny all from any to 127.0.0.0/8
$fwcmd add deny ip from 127.0.0.0/8 to any

# Engedélyezzük az Internetről hozzánk látogatóknak, hogy elérhessék
# a 22-es ill. a 80-as portokat. Így ez a példa kifejezetten az SSH
# (sshd) és HTTP (webszerver) típusú kapcsolatokat engedélyezi.
$fwcmd add allow tcp from any to me dst-port 22,80 in via $oif setup keep-state

# Engedélyezzük az ICMP csomagokat: vegyük ki a 8-as típust, ha nem
# szeretnénk a gépünket pingek által elérhetővé tenni.
$fwcmd add allow icmp from any to any via $oif icmp types 0,3,8,11,12

# Tiltsunk és naplózzunk minden mást.
$fwcmd add deny log ip from any to any
```

Most már van egy teljesen működőképes tűzfalunk, amely csak és kizárólag a 22-es, 80-es portokon enged kapcsolatot létesíteni, és minden egyéb próbálkozást naplóz. Így már nyugodtan újraindíthatjuk a rendszerünket, és ezt követően a tűzfalunk magától elindul és a hozzá tartozó szabályrendszer betöltődik. Ha bármilyen hibát találna benne vagy problémába ütközne a használata során, esetleg valamilyen építő jellegű javaslata van, kérem, keressen meg e-mailben!

## 6. Kérdések

1. “limit 500 reached on entry 2800”. Ilyen és ehhez hasonló hibaüzeneteket kapok, miután a számítógépem abbahagyja a szabályhoz tartozó eldobott csomagok naplózását. Működik még ilyenkor ea tűzfalam?

Ez csupán annyit jelent, hogy az adott szabályt elérte a hozzátartozó maximális naplóbejegyzést. A szabály maga még mindig aktív, viszont addig nem fog tudni naplózni, amíg nem töröljük valahogy a bejegyzésszámlálóját. Például így lehet törölni az említett számlálót:

```
# ipfw resetlog
```

Vagy úgy is elkerülhetjük ezt a hibaüzenetet, ha növeljük a szabályhoz tartozó naplóbejegyzések számát a rendszermag konfigurációs állományában, az IPFWALL\_VERBOSE\_LIMIT beállítás megváltoztatásával, a fentebb leírt módon. A rendszermag újrafordítása eacut;s a rendszer újraindítása nélkül is megváltoztatható ez a korlát, a net.inet.ip.fw.verbose\_limit sysctl(8) használatával.

2. Valami nem stimmel. Követtem a leírásban szereplő utasításokat pontról pontra, de kizártam magamat.

A leírás feltételezi, hogy *felhasználói ppp-t* használunk, és ezért a megadott szabályok a `tun0` (amely megfelel a `ppp(8)` (azaz *felhasználói ppp, user-ppp*) által létrehozott első kapcsolatnak) felületen keresztül működnek. A további kapcsolatok rendre a `tun1`, `tun2` stb. neveket használják.

Továbbá érdemes megjegyezni, hogy a `pppd(8)` ehelyett a `ppp0` felületet használja, így tehát ha a PPP-kapcsolatot a `pppd(8)`-al indítottuk el, akkor a `tun0` neveket mindenhol `ppp0` nevekre kell cserélni. Íme egy példa arra, hogyan írjuk át gyorsan a szabályainkat ilyen alakúra (az eredeti szabályokat pedig `fwrules_tun0` néven elmentjük):

```
% cd /etc/firewall
/etc/firewall% su
Password:
/etc/firewall# mv fwrules fwrules_tun0
/etc/firewall# cat fwrules_tun0 | sed s/tun0/ppp0/g > fwrules
```

Legkönnyebben úgy tudjuk kideríteni, hogy van `ppp(8)`-t vagy éppen `pppd(8)`-t használunk, hogy átnézzük az `ifconfig(8)` kimenetét, amikor már van aktív kapcsolatunk. Például, ha a kapcsolatot a `pppd(8)`-vel hoztuk létre, akkor valami ilyesmit kellene látnunk (csak a lényegét mutatjuk):

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

Másrészt viszont a `ppp(8)`-vel (vagyis *felhasználói ppp*-vel) létesített kapcsolatok esetén nagyjából ezt:

```
% ifconfig
(kimarad...)
ppp0: flags=8010<POINTOPOINT,MULTICAST> mtu 1500
(kimarad...)
tun0: flags=8051<UP,POINTOPOINT,RUNNING,MULTICAST> mtu 1524
        (IPv6 kimarad...)
        inet xxx.xxx.xxx.xxx --> xxx.xxx.xxx.xxx netmask 0xffffffff00
        Opened by PID xxxxx
(kimarad...)
```