Tűzfal és internetmegosztás OpenBSD-vel

Keresztes Ákos
xsak@c2.hu
Verzió: v0.1.20

Tartalomjegyzék

Módosítások:

Dátum:	*Verzió:*	*Módosítás:*
		1.0-ig nem kerül ide semmi

Még:

Kiegészítésre/hozzáadásra várnak a következő témák/lehetőségek. Aki tud ezek bármelyikében is akár csak egy mondattal segíteni, rögtön írjon!

- Kábelnetes elérés beállítása („©he110” ...)
- ~~Forgalom optimalizálás (feltöltéskor ne lassuljon le teljesen a letöltés) altq?~~
- Forgalommérés, statisztikák.
- Levelezőszerver a belső hálózatnak (imap?, pop3? Eléréssel).
- ~~BitTorrent anchor-ral~~
- Betörés-detektálás, ssh biztonságosabbá tétele.
- FTP csak aktív módban! Passzív korlátozottan.
- Proxy a hálózatnak (squid)?
- /etc/nat.conf-hoz az RDP portja 3389 (ha valaki belső hálón lévő WinXP-t akar távolról adminisztrálni) ((M. E. javaslata alapján))
-

Mi ez?

Leírás egy ADSL-kapcsolat megosztásáról szól automatikus IP-cím kiosztással és névfeloldással. Egy belső hálózat gépei érhetik el a tűzfalon keresztül viszonylag biztonságosan az internet szolgáltatásait.

E leírás egy házi megoldást, esetleg egy kis cég hálózatét ecseteli. Kérlek ne hagyd figyelmen kívül a **Figyelmeztetés!** szakaszban írottakat.

Figyelmeztetés!

Az ebben a dokumentumban leírtakat legjobb tudásom és eddigi tapasztalataim szerint próbáltam összeállítani, ennek ellenére hibákat, elírásokat, tévedéseket tartalmazhat. Ezekért s bármilyen más következményből eredő károkért, hátrányokért **felelősséget nem vállalok**.

Ha az itt leírtakat odafigyeléssel, kellő kritikával és józan ésszel olvasod, szerintem nem lesz problémád a saját hálózatodat/tűzfaladat beállítani.

Hiba, elírás vagy egyéb helytelenség esetén keress meg a következő e-mail címen: xsak@c2.hu

Megfontolások

Alapelv minden nyilvános gépen, hogy csak a szükséges programok és szolgáltatások fussanak a gépen, vagy akár feltelepítve legyenek. Minden egyes újabb program egy újabb kockázati tényezőt jelent.

Különösen fontos a biztonság egy eleve védelmi funkciót ellátó gépen, mint a mi tűzfalunk is. Vegyünk néhány alap megfontolást:

- Minél kevesebb szolgáltatás/program fusson,
- Minél kevesebb felhasználó legyen, s ők is minél kevesebb hozzáféréssel bírjanak,
- A felhasználók jelszava legyen kellően bonyolult, s rendszeresen cseréljék azokat! („A jó jelszó olyan, mint a fogkefe: naponta használjuk, gyakran cseréljük, és soha nem adjuk kölcsön senkinek!”)
- Legyen részletes naplózás, s naplókat ellenőrizni kell rendszeresen,
- Működjön egy betörés detektáló rendszer
- ...

OpenBSD

Az OpenBSD operációs rendszer egy ingyenes, több plattformos UNIX-szerű operációs rendszer, ami a 4.4BSD-n alapul. A projekt a következőkre helyezi a hangsúlyt: portabilitás (könnyű átírhatóság másik plattformra), sztenderdizálás, hibátlanság, megelőző biztonság és beépített titkosítással.

Az OpenBSD-t önkéntesek fejlesztik, s adományokból, CD-k eladásából tartják el magukat. Magyarországon csak régi CD-t lehet vásárolni FIXME :-(.

Miért OpenBSD-t használunk tűzfalnak?

Az OpenBSD-t a legbiztonságosabb megoldások között tartják számon a tűzfalak világában. Bár nem csak tűzfal, hanem egy teljes operációs rendszer, a leggyakoribb alkalmazása mégis az, hogy tűzfalként funkcionál. Kiszolgálónak inkább egy FreeBSD rendszert ajánlhatok, amely nagyon stabil és megbízható. Természetesen az OpenBSD is lehet kiszolgáló, futtathatunk rajta web-, ftp- vagy bármely unix alapú kiszolgáló szoftvert (sőt, desktopként is funkcionál).

Tűzfalunkon lehetőleg ne futtassunk semmit egyebet, hálózati kiszolgálásra egy újabb szervert állítsunk be. Persze az otthoni hálózatunknál költségtakarékos megoldás lehet egy gépen elhelyezni minden szükséges funkciót, ebben az esetben ez is megoldás.

Az OpenBSD fejlesztői büszkék arra, hogy 8 éves története alatt az alaprendszerben csupán egy, azaz 1 darab távolról kihasználható hibát találtak. Más rendszerekkel összevetve ez nagyon kimagasló biztonságot jelent, azt mutatja, hogy a fejlesztők nagyon odafigyelnek a biztonsági auditra. A biztonsági elemek eleve működnek már az alap telepítésnél, szinte semmit sem kell kézileg átállítanunk, nem kell a biztonság növeléséhez patchelni a kernelt vagy biztonsági szoftvereket telepítenünk. Az OpenBSD "alapból nagyon biztonságos". De nemcsak telepítésre kerülnek ezek az elemek, hanem a rendszer naponta ellenőrzi is, hogy minden megfelelő-e, nincs-e valami biztonsági rés. Érdemes belenézni az **/etc/security** szkriptbe, amely ezt az ellenőrzést hajtja végre és elolvasni a "security manualt" (man security), a szkript rövid leírását.

Ha más operációs rendszert, mondjuk Linuxot használunk, sokkal több munkánk lesz a tűzfal megerősítésére, könnyen hibázhatunk és sokkal nehezebben átlátható rendszerünk lesz. Ha az OpenBSD-t választjuk, a biztonsági kérdéseket ráhagyhatjuk a fejlesztőkre, míg elegendő ismeretünk nem lesz a finomhangoláshoz. Rendszerünk így is megnyugtatóan biztonságos lesz. Végül, de nem utolsósorban, az OpenBSD 100%-osan szabad, ingyenes szoftver.

Eszközök

A feladat tulajdonképpen egy kis hálózat építése. Ehhez használhatunk új, minőségi eszközöket is, de például egy otthoni változathoz nem szükséges a csúcstechnika. Egy elérhető 384/64 sebességű ADSL vonal esetén például aligha a hardverek határozzák meg az „élményt” (persze nem értem ide a nagyon szélsőséges eseteket). Nálam majdnem minden hardver kapott vagy használtan vásárolt.

1 Tűzfal számítógép:

A megosztáshoz egy régi, más által kidobásra ítélt számítógépet használok (kösz Miki!). Pentium I. processzor, 188 Mhz-re „felhúzva”. Régi EDO-RAM-okból 64 MB-ot sikerült belerakni, ami bőven elegendő. Merevlemeznek két kisebb winchestert alkalmazok.

Szerintem az alábbi konfiguráció alá hardverben ne menjünk:

CPU:	Pentium I. - 100MHz
Memória:	32 MB
HDD:	800 MB

A gyors CPU és a sok memória fontos a programok fordítása miatt. Az OpnBSD-ben (mint a BSD-kben általában) sok programot kell forrásból fordítani. A memória a tűzfal működése közben is szempont, főként, ha sok szabályunk van.

A merevlemezt főként az operációs rendszer foglalja el, adatot itt nem nagyon tárolunk. Az általam javasolt minimális 800 MB úgy elég, ha a forrásokat máshol (pl. NFS megosztáson) tároljuk. Ha mindent ezen a gépen kívánunk tartani, akkor 2GB a szükséges.

Hálózati kártyának szinte bármilyen jó, ha nem a legújabb, akkor szinte biztos, hogy fel lehet éleszteni OpenBSD alatt. Amin Realtek chip van, szinte biztosan alkalmazható.

2 Hálózati elosztó eszköz:

A belső hálózat gépeit nálam egy egyszerű, nem menedzselhető, 8 portos switch kapcsolja össze, viszonylag olcsón beszerezhető. Ennél a megoldásnál ún. egyenes kötésű kábeleket használjunk.

Ha csak egy gépet kötünk a tűzfalra, akkor a tűzfalat és a kliens gépet egy keresztkábellel csatlakoztassuk egymáshoz. Az ADSL modem és a megosztó gép között szintén keresztkábel a megfelelő.

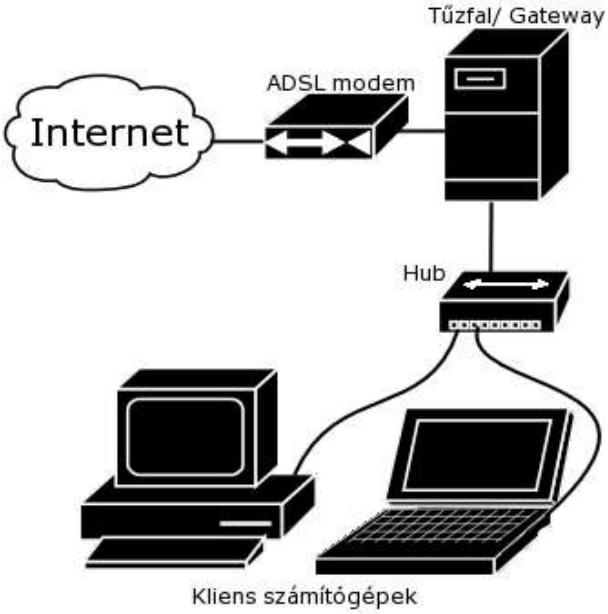
3 Kliensek:

Szinte bármi lehet, ami a DHCP-t ismeri, s alkalmazza. Több Linux fut a gépemen, mindegyikkel rögtön tudtam internetezni. Amivel próbáltam: FreeBSD, FreeSBIE, Windows (XP 98), Knoppix, Sulix, SuSE, Mandrake, Slackware, Gentoo, SLAX.

A klienseken beállíthatunk fix IP címet is, de ezzel kockáztatjuk azt, hogy egy DHCP-s géppel ütközni fog. Ha mégis így teszünk, a tűzfalgép belső IP címét állítsuk be alapértelmezett átjárónak (default gateway) és névkiszolgálónak (DNS server).

Telepítés

1 Elrendezés



Az internetet az ADSL modem hozza be a házba, amit egy keresztkábellel bekötünk a tűzfalgépünk egyik hálózati csatolójába. A tűzfal másik csatolóját vagy közvetlenül az elosztó eszközünkre (hub/switch/...) dugjuk, vagy ha csak egy gépet akarunk rákötni, akkor ahhoz csatlakoztatjuk (keresztekábellel).

A tűzfal hálózati címfordítást (NAT) végez, s ezen kívül csomagszűrést (packet filtering). A NAT-olás azt jelenti FIXME, hogy a belső hálózat gépei által az internet felé küldött csomagok forráscímét a NAT-ot végző eszköz kicseréli a saját publikus címére, s továbbküldi a célállomás felé, majd az erre érkező választ továbbítja a az eredeti gépnek. Részletesebben lásd a HUP Wiki NAT címszavát. A csomagszűrés során a tűzfal a hálózati forgalmat szabályozza. A szűrés egy-egy interfészen (hálózati kártyán) történik, s az éppen vizsgált csomagot vagy átengedi, vagy eldobja. Az átengedés/eldobás eldöntésének alapja az aktuális csomag forrás és cél IP címe, protokollja és portszáma. Az OpenBSD pf tűzfala ún. állapottartó csomagszűrést végez, ami azt jelenti, hogy a tűzfal figyelemmel kíséri a hálózati kapcsolatok állapotát. Az ilyen tűzfal úgy van beállítva, hogy tudja az egyes kapcsolati típusoknál, melyek azok az érvényes csomagok, amik jogosultak az áthaladásra.

1.Operációs rendszer telepítése

Az OpenBSD telepítése sokféle módon történhet.

1 Vásárolt CD telepítőkészlettel. Ez lenne az üdvözítő megoldás, de tapasztalataim szerint csak elég régi lemezeket lehet Magyarországon beszerezni.

2 Interneten keresztül, boot floppy-val. Erről egy nagyon jó leírást olvashatunk a http://www.hup.hu/modules.php?name=News&file=article&sid=5926&mode=&order=0&thold=0 címen.

3 Saját telepítő CD-vel. Erről is olvasható egy kiváló leírás itt:

http://www.hup.hu/modules.php?name=News&file=article&sid=5959&mode=&order=0&thold=0. Én ezt tárgyalom.

Telepítő CD létrehozása

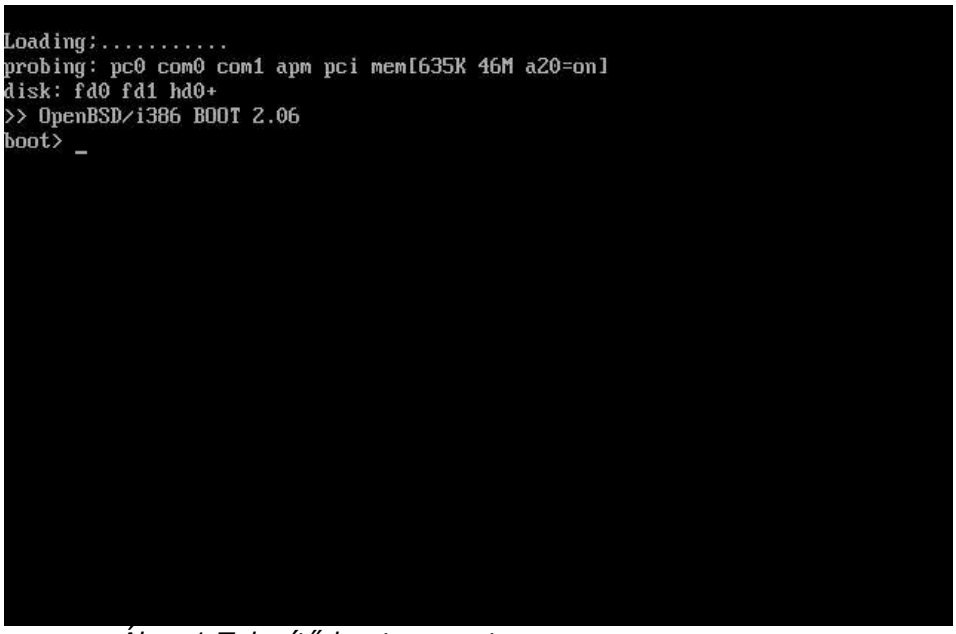
1. A saját telepítő CD létrehozásához hozzunk létre egy ideiglenes könyvtárat valahol a számítógépünkön (Pl. OpenBSD).
2. Ez alatt hozzunk létre egy „3.5” nevű alkönyvtárat.
3. A 3.5 nevű könyvtárba töltsük le az ftp kiszolgálóról az ottani 3.5 alatti „i386” és „tools” könyvtárak tartalmát.
4. Töltsük még le az XF4.tgz, ports.tgz, src.tgz, sys.tgz forrásfájlokat is ide.
5. Ha akarjuk, az egyéb fenn lévő szövegfájlokat is idetehetjük.
6. A helyi „OpenBSD” könyvtárunkban adjuk ki a következő parancsot (a ponttal a végén!):

```
mkisofs -o ../obsd35.iso -R -J -V "OpenBSD 3.5 CD" -b 3.5/i386/cdrom35.fs -c
"boot.catalog" -A "OpenBSD 3.5 CD" .
```

Ezzel kész az indítható CD image, amit egy lemezre felírva, rögtön kezdhetjük a rendszer telepítését.

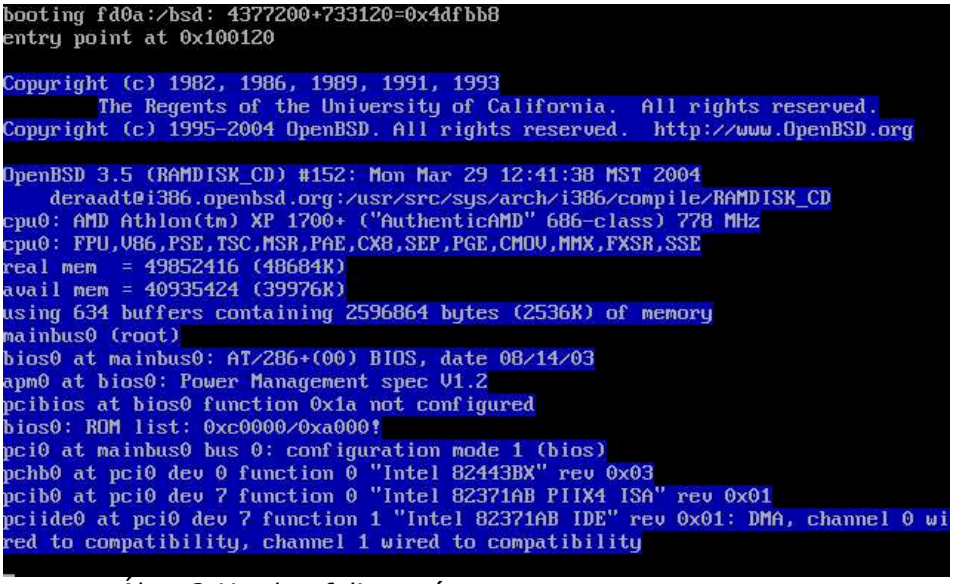
Telepítés menete

Behelyezzük a CD-t a meghajtóba, s arról indítjuk a gépet (BIOS-ban átállítandó).



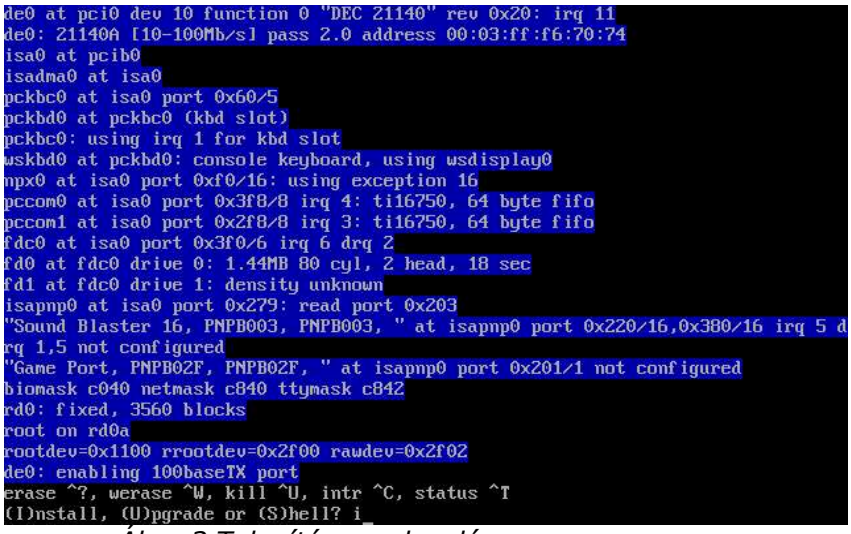
Ábra 1 Telepítő boot prompt

Bejön a boot prompt, ahol vagy megvárjuk a rendszer indulását, vagy nyomunk egy Entert.



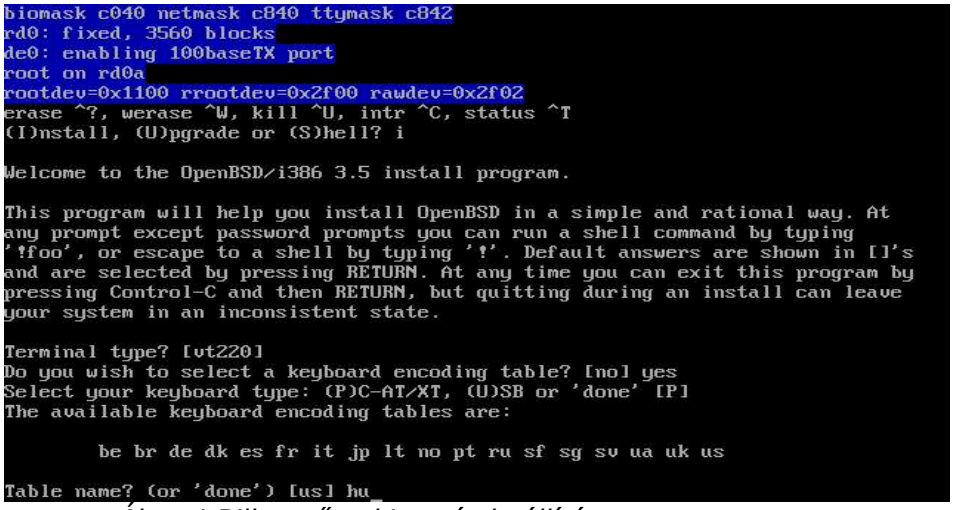
Ábra 2 Hardverfelismerés

A kernel ezután megpróbálja felismerni a számítógép hardverelemeit. Ha ezalatt megáll a folyamat, s sokáig nem lép tovább, akkor indítsuk újra a számítógépet, s kapcsoljuk ki az APCI-t a BIOS-ban, így boot-oljunk. (A kék feliratok a kernel kimenetét jelölik)



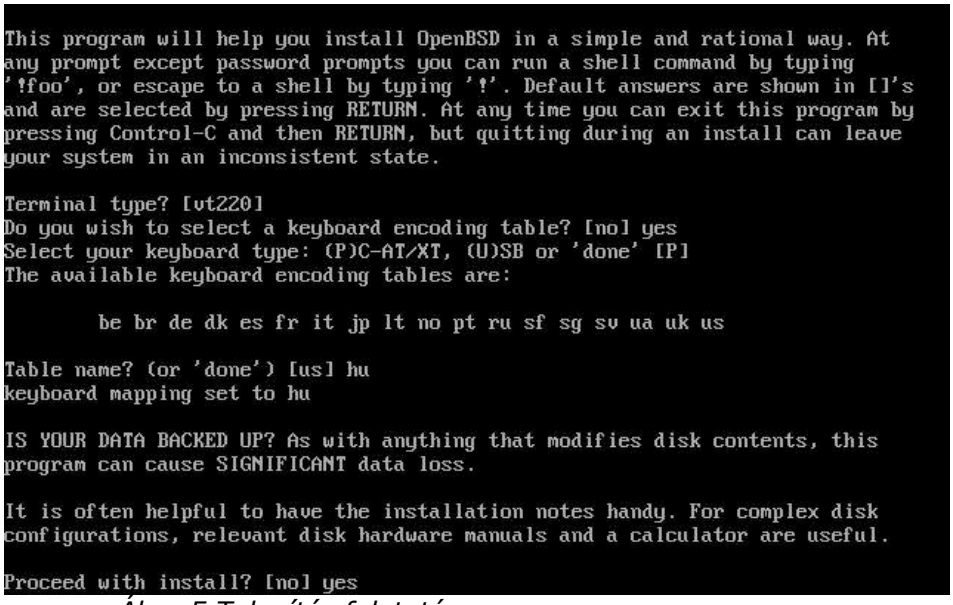
Ábra 3 Telepítés megkezdése

Rögtön kiválaszthatjuk, hogy most telepíteni, frissíteni szeretnénk, vagy csak egy shell-t szeretnénk egyéb helyreállító feladatok ellátására. Írjunk „i”-t, a telepítés megkezdéséhez.



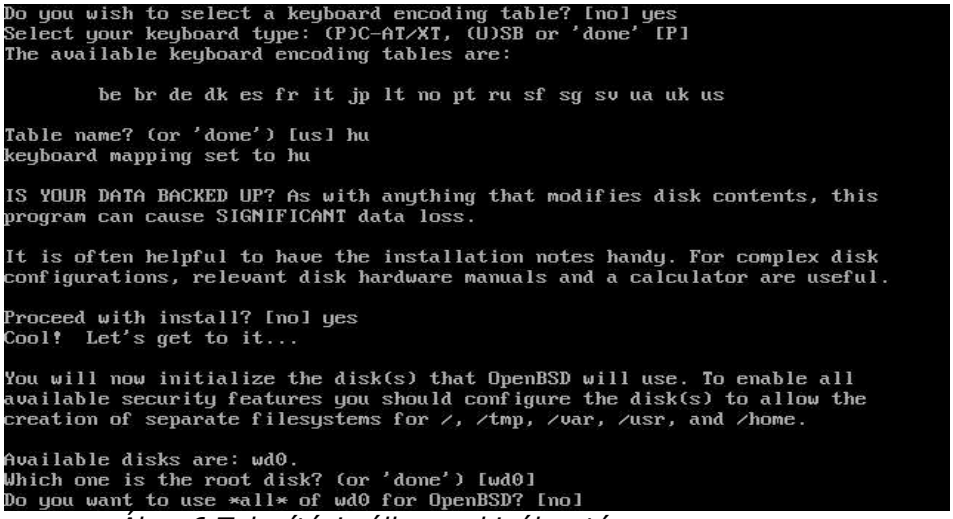
Ábra 4 Billentyűzetkiosztás beállítása

Ezután a terminál típusát állíthatjuk be, fogadjuk el az alapértelmezett vt220-at egy Enter lenyomásával. Majd rákérdez arra, hogy akarjuk-e átállítani a billentyűzetkiosztást, „yes”-t válaszolva megadható a billentyűzet típusa és kiosztása. Bár a magyar kiosztás nincs feltüntetve, de a „hu” kód beírásával használható.



Ábra 5 Telepítés folytatása

A biztonsági mentések fontosságának hangsúlyozása után rákérdez, hogy folytatható-e a telepítés. Válaszoljunk „yes”-szel.



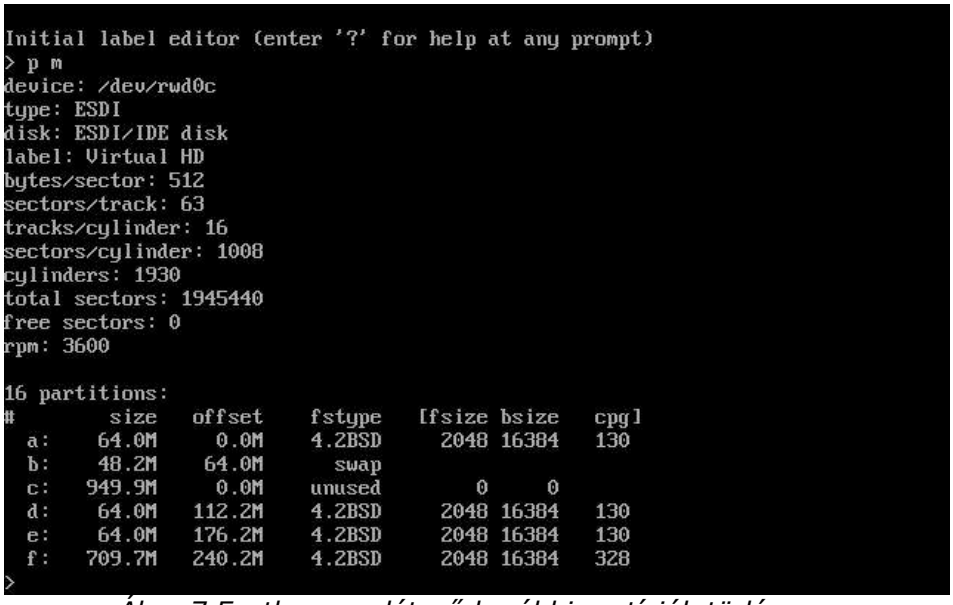
Ábra 6 Telepítési céllemez kiválasztása

Most jön a particionálás. A BSD rendszerek kicsit másként jelölik a hardverelemeket, s a merevlemezeken a partíciókat (slice-okat). Erről mindenképpen előzetesen tájékozódjunk pl. itt: http://www.bsd.hu/dokumentacio/bevezetes/slice/view. A szükséges hely körülbelül 1 GB, noha ez persze leszorítható kisebbre, de 1,5 GB felett már kényelmesen elférhetünk. **Mindenki a saját igényeihez és lehetőségeihez igazítsa a lemez felosztását!**

Ha az egész merevlemezt az OpenBSD-nek szánjuk, mondjunk itt „yes”-t, egyébként a „no”-ra nyomhatunk Entert. A felosztást a következők szerint képzelem (Persze ha több a hely, nem baj :-):

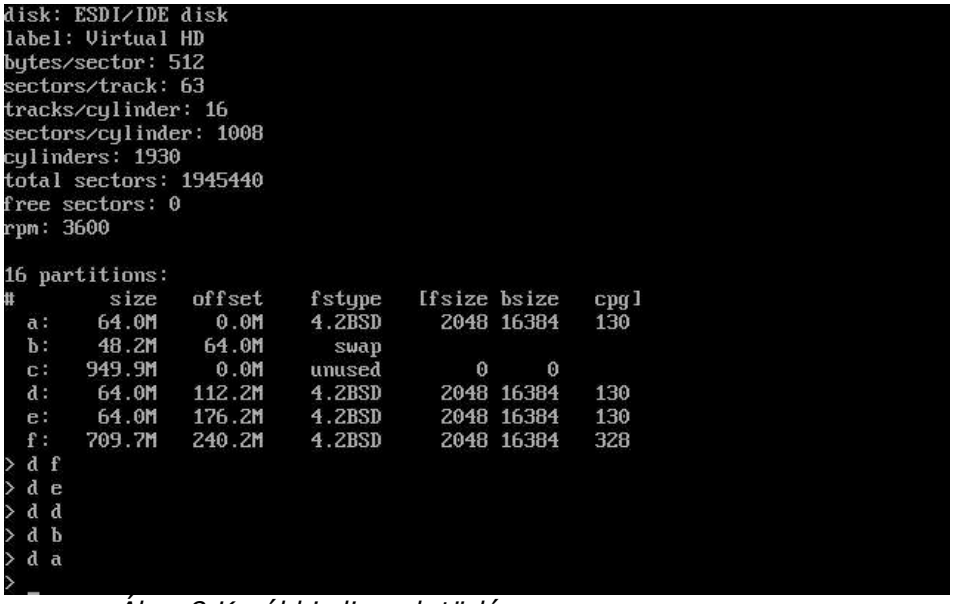
slice	***méret***	***Mountpoint***
a	64M-256M	/
b	32M-256M	swap
d	64M-256M	/var
e	64M-256M	/tmp
f	400M -	/usr

Ehhez a telepítő CD-n található particionáló programot használjuk. Használata nem túl bonyolult, legfontosabb parancsai: „a” = slice hozzáadása, „d” = slice törlése, „p” = kiíratás, „p m” = kiíratás MB-ban, „h” = súgó.



Ábra 7 Esetlegesen létező korábbi partíciók törlése

A képen látható slice-ok egy korábbi OpenBSD telepítésből származnak, ezeket először töröljük. A „c” slice az egész lemezt jelöli, ezt nem bántjuk.



Ábra 8 Korábbi slice-ok törlése

A korábbi slice-okat a „d” paranccsal töröljük sorban, pl.: „d f”.

```
  e:    64.0M   176.2M    4.2BSD     2048 16384   130
  f:   709.7M   240.2M    4.2BSD     2048 16384   328
> d f
> d e
> d d
> d b
> d a
> p m
device: /dev/rwd0c
type: ESDI
disk: ESDI/IDE disk
label: Virtual HD
bytes/sector: 512
sectors/track: 63
tracks/cylinder: 16
sectors/cylinder: 1008
cylinders: 1930
total sectors: 1945440
free sectors: 1945377
rpm: 3600

16 partitions:
#        size   offset    fstype   [fsize bsize   cpg]
  c:   949.9M     0.0M    unused        0     0
> _
```

Ábra 9 Nincs semmi a lemezen

Most már nincs semmi a lemezen, hozzákezdhetünk a kívánt partíciók létrehozásához.

```
> d a
> p m
device: /dev/rwd0c
type: ESDI
disk: ESDI/IDE disk
label: Virtual HD
bytes/sector: 512
sectors/track: 63
tracks/cylinder: 16
sectors/cylinder: 1008
cylinders: 1930
total sectors: 1945440
free sectors: 1945377
rpm: 3600

16 partitions:
#        size   offset    fstype   [fsize bsize   cpg]
  c:   949.9M     0.0M    unused        0     0
> a a
offset: [63]
size: [1945377] 64m
Rounding to nearest cylinder: 130977
FS type: [4.2BSD]
mount point: [none] /
> _
```

Ábra 10 Gyökér slice létrehozása

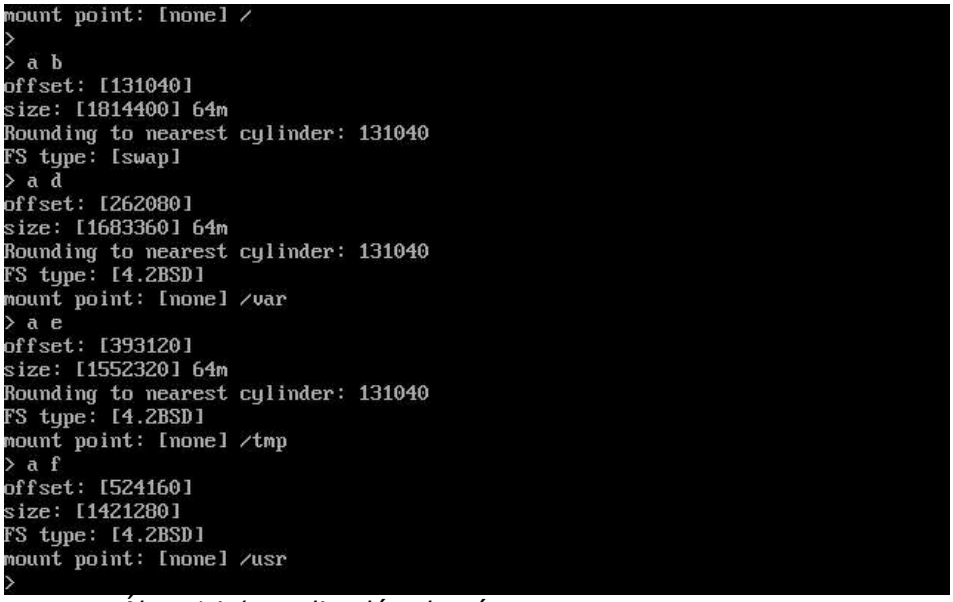
```
free sectors: 1945377
rpm: 3600

16 partitions:
#        size   offset    fstype   [fsize bsize   cpg]
  c:   949.9M     0.0M    unused        0     0
> a a
offset: [63]
size: [1945377] 64m
Rounding to nearest cylinder: 130977
FS type: [4.2BSD]
mount point: [none] /
>
> a b
offset: [131040]
size: [1814400] 64m
Rounding to nearest cylinder: 131040
FS type: [swap]
> a d
offset: [262080]
size: [1683360] 64m
Rounding to nearest cylinder: 131040
FS type: [4.2BSD]
mount point: [none] /var
>
```

Ábra 12 /var slice létrehozása

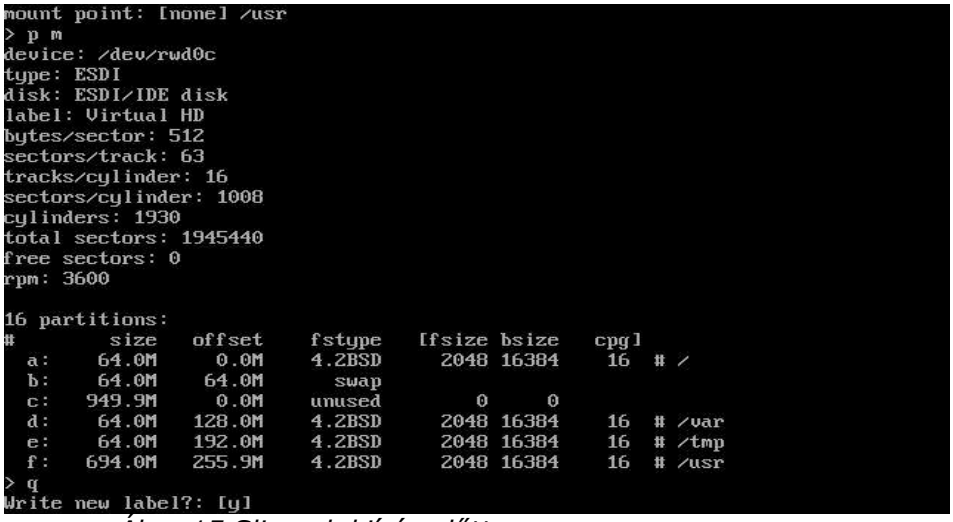
```
> a a
offset: [63]
size: [1945377] 64m
Rounding to nearest cylinder: 130977
FS type: [4.2BSD]
mount point: [none] /
>
> a b
offset: [131040]
size: [1814400] 64m
Rounding to nearest cylinder: 131040
FS type: [swap]
> a d
offset: [262080]
size: [1683360] 64m
Rounding to nearest cylinder: 131040
FS type: [4.2BSD]
mount point: [none] /var
> a e
offset: [393120]
size: [1552320] 64m
Rounding to nearest cylinder: 131040
FS type: [4.2BSD]
mount point: [none] /tmp
>
```

Ábra 13 /tmp slice létrehozása



Ábra 14 /usr slice létrehozása

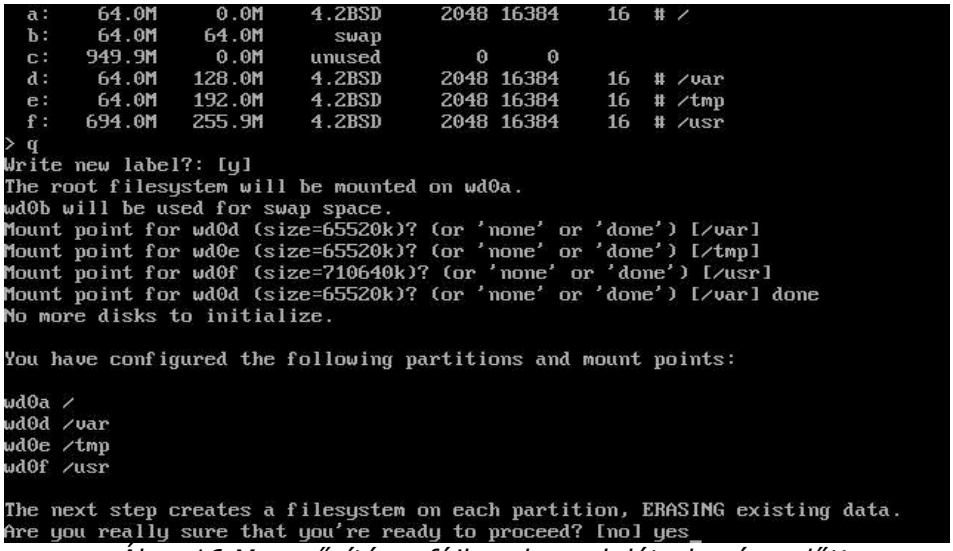
Ellenőrizzük még egyszer mit csináltunk:



Ábra 15 Slice-ok kiírás előtt

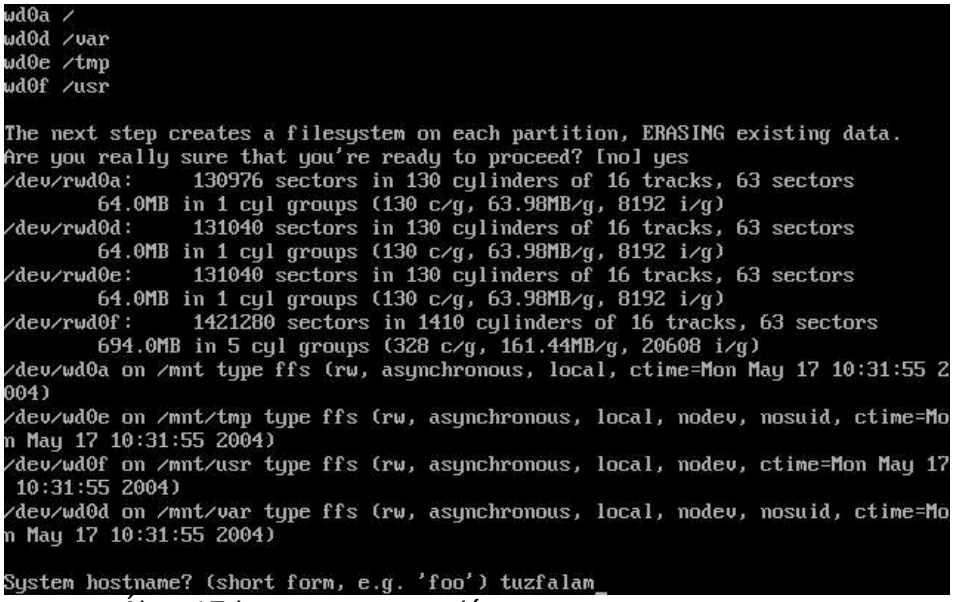
A „p m” paranccsal nézzük meg, hogyan sikerült a felosztás, s végezzük el az esetleges javításokat, ha hibát találunk. A „q”-val léphetünk ki, amire a rendszer rákérdez, hogy kiírja-e az új lemezcímkét.

Ezután még felülbírálhatjuk az egyes slice-ok csatolási pontjait (mount point), ha rendben találjuk, írjuk be a „done” varázsszót.



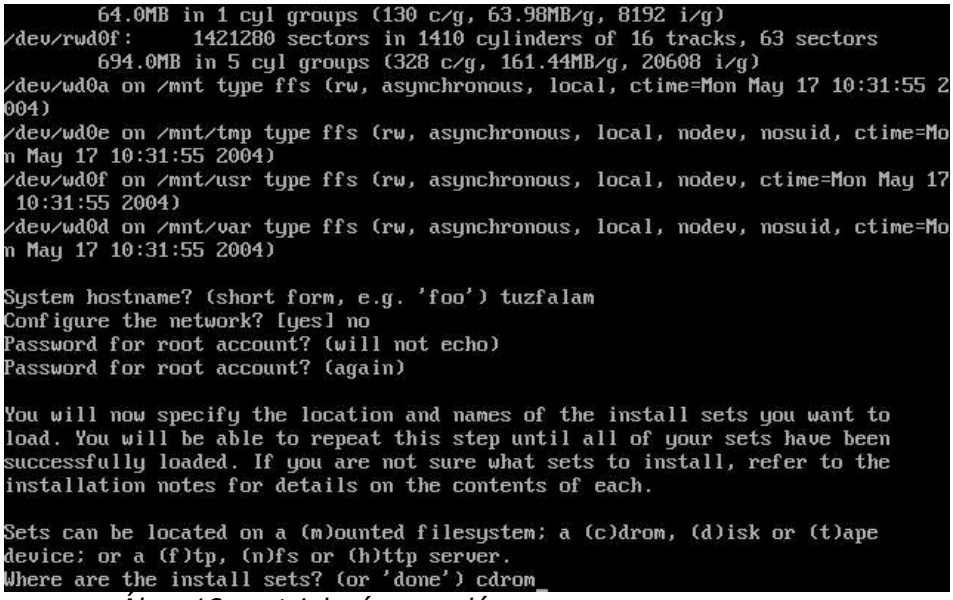
Ábra 16 Megerősítés a fájlrendszerek létrehozása előtt

A változtatások valós végrehajtása előtt még egy megerősítést kell tennünk. Írjuk be: „yes”!



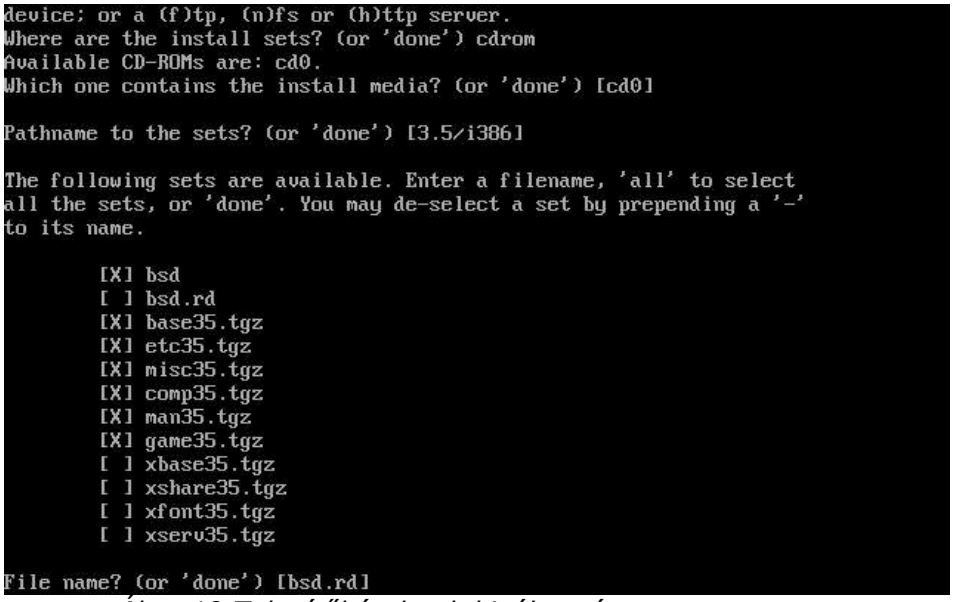
Ábra 17 hostname megadása

A lemez inicializálása után a számítógépünk hostnevét adhatjuk meg. Válasszunk egy nekünk tetsző nevet.



Ábra 18 root jelszó megadása

A névadó után adjunk egy megjegyezhető, de kellően bonyolult jelszót a rendszergazdának, miután rögtön kiválaszthatjuk, hogy honnan kívánjuk a telepítési készleteket felhasználni. Mi most a CD-ROM-ról telepítünk, hát írjuk be: „cdrom”.

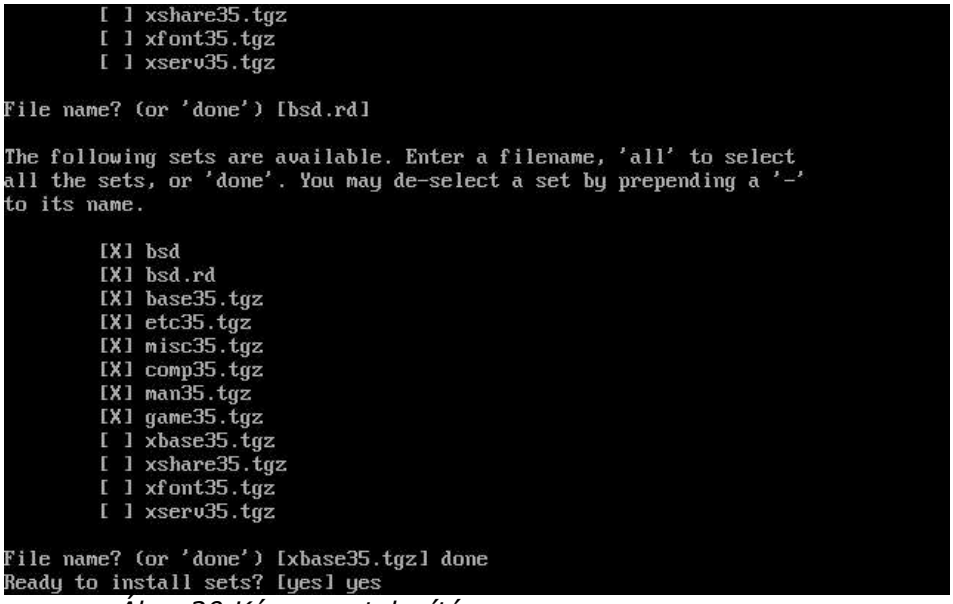


Ábra 19 Telepítőkészletek kiválasztása

Megadjuk a telepítő CD helyét, ami általában „cd0” (felajánlja), s ezen belül is a készletek könyvtárát (3.5/i386). Az itt látható készletekből választhatunk. Beírjuk a hozzáadandó készlet nevét, s nyomunk erre egy Entert, ha kivenni szeretnénk, akkor tegyünk elé egy mínusz jelet. Ha kész, írjuk be a „done” lezáró szót.

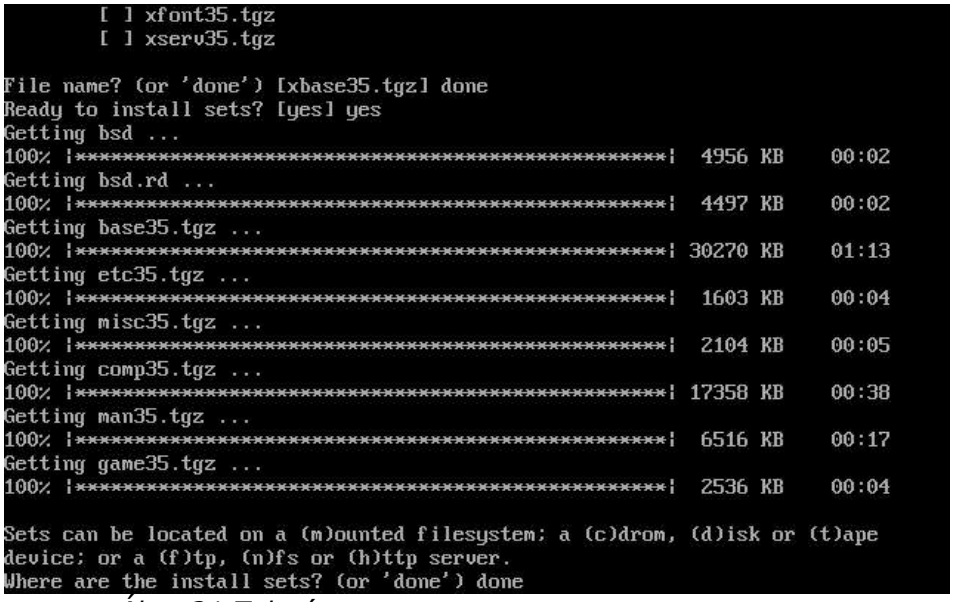
Javaslom a „bsd.rd” készlet telepítését, ahol az „rd” jelentése ramdisk. Ha ezzel a kernellel indítjuk a gépet („boot /bsd.rd” parancsa a boot promptnál), akkor csak memóriából

fut egy alap BSD rendszer, amivel a későbbiekben frissítési/adminisztrációs feladatokat lehetséges elvégezni.



Ábra 20 Készen a telepítésre

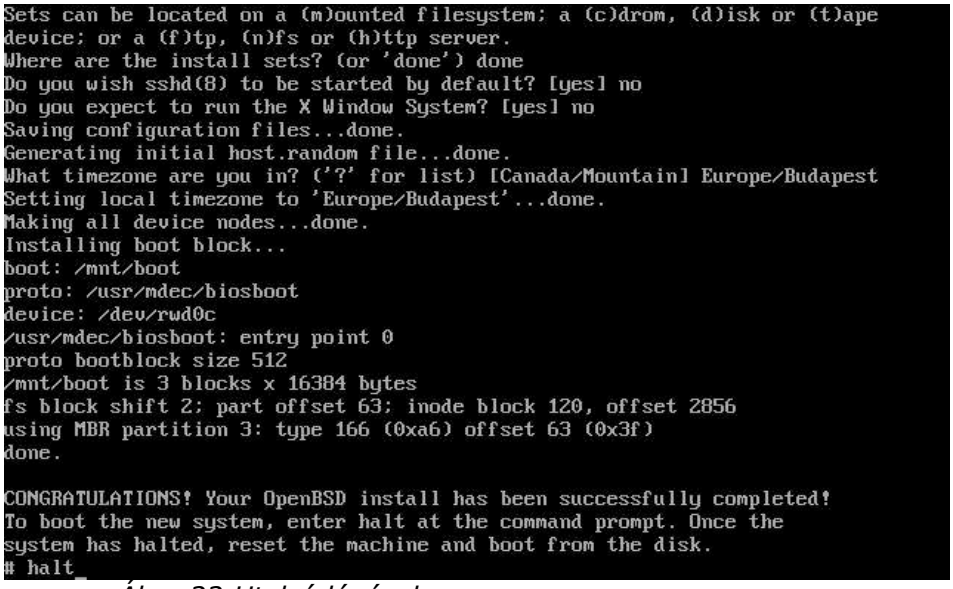
A készletek kiválasztása után kapunk még egy megerősítő kérdést, hogy egészen biztosan készen állunk a telepítés megkezdésére.



Ábra 21 Telepítve

A telepítés elindul, s ideális esetben sikeresen lefut. Ezután még választhatunk másik forrást is, ha például egyéni készletünk is van (lásd „Egyéni, vagy sok hasonló rendszer

telepítése” a Kapcsolódó dokumentumoknál). Ha nem telepítünk máshonnan, írjuk be: „done”

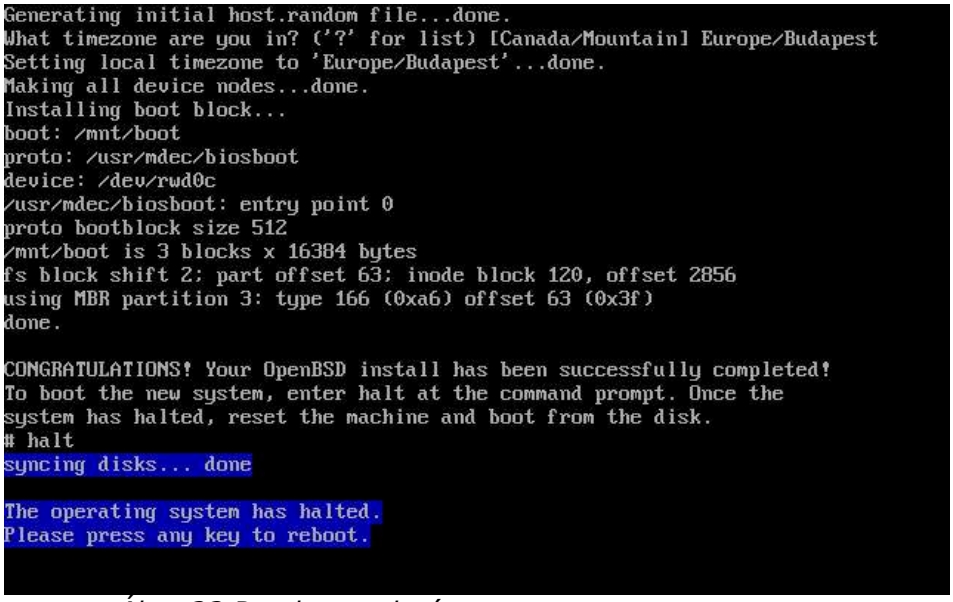


Ábra 22 Utolsó lépések

Eztán a telepítő megkérdezi, hogy szeretnénk-e, hogy az ssh démon elinduljon a gép indulásakor. Egyelőre ezt nem engedem, de adott esetben fontos lehet. Az Xfree86-ot nem várható, hogy használjuk, hisz tűzfalat telepítünk, ahol ez biztonsági kockázatot jelentene. Mindkét kérdésre „no”-val válaszolunk.

Néhány beállítás után még rákérdez az időzóna beállításra, amire nekünk „Europe/Budapest” válasszal célszerű reagálni.

Ezek után elkészülnek az eszközeink node-jai (ez elég sokáig tart lassú gépen), majd a rendszerindítás kerül elrendezésre. Ezzel végeztünk is az operációs rendszer telepítésével, újraindíthatjuk a gépet.



Ábra 23 Rendszer telepítve

A „halt" parancsra leáll a gép, ám újraindításkor ellenőrizzük, hogy ezentúl ne a CD-ről, hanem a vincseszterről induljon a számítógép. Ha a gép mások által elérhető helyen van, állítsunk be jelszót a BIOS-ban is.

1 Beállítás

1 Operációs rendszer beállítása

Egy picit növelhetjük a gépünk biztonságát, ha bizonyos fájlrendszereket speciális módon csatolunk be. A „**nosuid**" jelzőt (ami letiltja a setuid és setgid bitet érvényre jutni) beállíthatjuk a /var, /tmp, /home fájlrendszerekre (azaz a / és a /usr kivételével mindegyikre). A „**nodev**" jelző (ami nem engedi a fájlrendszeren a speciális karakter vagy blokk eszközök létét) mehet a gyökéren kívül mindegyikre. A /tmp pedig kapjon egy „**noexec**" jelzőt (ami tiltja a binárisok futtatását a fájlrendszeren).

Ezek után nálam a /etc/fstab fájl így néz ki:

/etc/fstab

```
/dev/wd0a / ffs rw 1 1
/dev/wd1f /home ffs rw,nodev,nosuid 1 2
/dev/wd1e /tmp ffs rw,nodev,nosuid,noexec 1 2
/dev/wd0d /usr ffs rw,nodev 1 2
/dev/wd1d /var ffs rw,nodev,nosuid 1 2

/dev/cd0a /mnt/cdrom cd9660 ro,noauto 0 0
```

A beállítások a következő becsatoláskor (újraindításkor) érvényre jut.

2 Helyi hálózat beállítása

A helyi hálózati kártya beállításához legelőször is tudni kell, a helyi hálózatra csatlakozó kártyát milyen néven ismeri az OpenBSD. A BSD-knél nem mindig ugyanaz az ethernet kártya neve, mint pl. Linuxban (eth0). Ha egy egyszerű, régi kártyát tettünk a gépbe,

akkor legvalószínűbb, hogy „rl0” vagy „ne3”/”ne4” néven ismerte föl. Nézzük végig a „dmesg” parancs kimenetét, ebből kiderül a kártyáink neve. Én ezt látom:

```
ne3 at pci0 dev 9 function 0 "Realtek 8029" rev 0x00: irq 11
ne3: address 00:e0:29:18:7c:87
ne4 at pci0 dev 10 function 0 "Realtek 8029" rev 0x00: irq 10
ne4: address 00:c0:26:ef:3c:e4
```

Tehát nálam van egy „ne3” és egy „ne4” nevű hálózati csatoló. Próbálgatással könnyen kitalálható, melyik melyik. Nálam az „ne3” lett a belső hálózati csatoló, hát ezt állítom be itt.

Ezután el kell dönteni, hogy milyen belső címeket használjunk kicsiny LAN-unkon. Legyen ez mondjuk egy C osztálybeli 192.168.0.0/24-es hálózat-tartomány (ami a 192.168.0.1-192.168.0.254 IP-cím tartomány jelenti). Természetesen használhatunk A osztályú címeket is (10.0.0.0/8), de talán ez ágyúval verébre tipikus esete lenne (Alternatív megoldás a kettő vegyítése lehet, tehát mondjuk 10.102.55.0/24 C osztályú hálózat alkalmazása (címek 10.102.55.1-től 10.102.55.254-ig)) 10-es címekkel.

Tehát az „ne3” hálókártya felélesztéséhez írjuk be a következőt a /etc/hostname.ne3 fájlba:

/etc/hostname.ne3

```
inet 192.168.0.1 255.255.255.0 NONE
```

A következő újraindításkor ez a beállítás érvényre jut, tűzfalunkat elérhetjük ezen az IP címen. Ha rögtön és kézzel szeretnénk elérni ezt, gépeljük be:

```
ifconfig ne3 inet 192.168.0.1 netmask 255.255.255.0 up
```

3 ADSL kapcsolat beállítása

Az ADSL egy furcsa jószág, de gyorsan beállítjuk. Itt is a modemes kapcsolatoknál megszokott ppp démont használjuk. A ppp kapcsolat egy bizonyos virtuális interfészen keresztül zajlik, aminek neve „**tun0**”, ez megfelel a Linux alatti **ppp0** interfésznek.

A másik hálózati csatolót „up”-ra állítjuk, hogy induláskor a kártya aktív legyen, de ne kapjon IP címet. Írjuk be a /etc/hostname.ne4 fájlba az „up” szócskát:

/etc/hostname.ne4

```
up
```

A /etc/ppp/ppp.conf fájlt ehhez hasonlóan kell kitölteni:

/etc/ppp/ppp.conf

```
default:
 set log phase Chat LCP IPCP CCP tun command
 set redial 15 0
 set reconnect 15 10000

pppoe:
 set device "!/usr/sbin/pppoe -i ne4"
 disable acfcomp protocomp
 deny acfcomp
 set mtu max 1492
 set speed sync
 enable lqr
```

```
 set lqrperiod 5
 set cd 5
 set dial
 set login
 set timeout 0
 set authname user@szolgaltato.hu
 set authkey **********
 add! default HISADDR
 enable dns
 enable mssfixup
```

A „set authname” illetve a „set authkey” sorokba rendre a felhasználónevet és a jelszót kell beírni. Állítsuk a fájl jogosultságait csak root által olvashatóvá:

```
chown root:wheel /etc/ppp/ppp.conf
chmod 600 /etc/ppp/ppp.conf
```

Az ADSL kapcsolatot indítani a „**ppp -ddial pppoe**” paranccsal tudjuk. A kapcsolat létesítéséhez és lebontásához inkább használjuk ezt a két indítószkriptet (én a /usr/local/sbin könyvtárba tettem őket, de tehetjük őket más, útvonalban szereplő könyvtárba is):

/usr/local/sbin/adsl-start

```
#! /bin/sh
echo -n "Trying to establish PPPoE ADSL"; ppp -ddial pppoe
for i in 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0; do
        sleep 3
        echo -n "."
        if /usr/local/sbin/adsl-status > /dev/null ; then
                break
        fi
done
echo
```

/usr/local/sbin/adsl-stop

```
#! /bin/sh
echo -n "Stopping ADSL connection... ( "
ppppid=`ps ax | grep ppp | grep -v grep | awk '{ print $1 }'`
for i in $ppppid;
do
   echo -n "$i "
   kill $i
done
echo ")"
ifconfig tun0 delete
```

A kapcsolat állapotáról ezzel győződjünk meg:

/usr/local/sbin/adsl-status

```
#! /bin/sh

IP=$(/sbin/ifconfig tun0 | awk '/netmask/{print $2}')

if [ -z "$IP" ]; then
        echo "ADSL link is down."
        exit 1
else
        echo "ADSL is up. IP address is $IP"
```

```
        exit 0
fi
```

Persze használat előtt mindegyik szkriptet tegyük futtathatóvá, miután begépeltük őket:

```
chmod 744 /usr/local/sbin/adsl-*
```

Próbáljuk ki őket! Kapcsolódjunk az ADSL-lel, pingeljünk egy hosztot (pl. www.index.hu) nézzük meg az adsl-status parancs kiemenetét, majd bonthatjuk a vonalat.

A **/etc/ppp/ppp.linkup** és **/etc/ppp/ppp.linkdown** szkriptek a kapcsolat felépültekor illetve lebontásakor futnak le. Formátuma kicsit különös. Ezekbe még írunk dolgokat...

Ha azt szeretnénk, hogy az ADSL kapcsolat a gép indulásakor létrejöjjön, egészítsük ki a **/etc/rc.local** fájlt ehhez hasonló módon:

```
...

# Starting ADSL connection...
echo "Starting ADSL connection..."
  ifconfig ne4 up
#  route flush
  ppp -ddial pppoe
echo "Done."
```

4 Névszerver beállítása

Itt a cél egy olyan névszerver beállítása, ami belső hálózatunk számára a külső címeket feloldja, illetve aminek segítségével a belső hálózatunkon lévő gépeket is névvel érhetjük el. Belső zónánk neve legyen mondjuk „itthon.hu” (ne felejtsük, ez csak a belső hálózaton élő tartománynév). Megadunk néhány belső gépnevet, akiket majd a hálózati kártyájuk MAC címe alapján azonosítunk, s eszerint adunk IP címet a következő DHCP-t leíró rész alapján. Ezt az egész procedúrát célszerű a következő, „DHCP beállítása” szakasszal szervesen együtt végezni.

Az OpenBSD alapértelmezetten tartalmaz egy névkiszolgálót, a BIND egy verzióját, amelyet „jail”-ben, bebörtönözve futtat a „/var/named” könyvtár alatt a „named” felhasználó fiókjával. Ez biztonsági megfontolások miatt fontos. Létrehozunk egy belső zónát, mondjuk „itthon.hu” domain végződéssel, tehát gépeink neve ehhez hasonlóan fog alakulni: jani.itthon.hu, feri.itthon.hu, mari.itthon.hu, s ezen neveket csak a hálózatunkon belülről érhetjük el ilyen néven.

1 named.boot

Kezdjük a named indítófájljával, ami a named.boot. Ebben meghatározzuk a domainünk úgynevezett névfeloldási zónáit, reverse (visszafelé feloldás) zónáit, a helyi gép és egyéb speciális zónákat.

/var/named/named.boot

```
; FILE: /var/named/named.boot
directory /var/named
; Type  Domain                          Data file
```

```
primary itthon.hu                       namedb/itthon.hu.zone
primary 0.168.192.in-addr.arpa          namedb/itthon.hu.rev
primary localhost                       namedb/localhost.zone
primary 0.0.127.in-addr.arpa            namedb/localhost.rev
primary 0.in-addr.arpa                  namedb/all-zero.rev
primary 255.in-addr.arpa                namedb/all-one.rev
cache   .                               namedb/root.cache

; Static IP number:
; primary 192.168.0.1

; my parent nameservers:
forwarders                          194.149.0.157 194.149.0.156
```

A „forwarders” sorba a saját internetszolgáltatónk által megadott DNS kiszolgálók IP címeit adjuk meg!!!

A gyökér névszervereket lementjük a **/var/named/namedb/root.cache fájlba**. Ez elérhető itt: ftp://ftp.internic.net/domain/named.cache. Érdemes időnként ezt a fájlt ellenőrizni, s újabb verzió esetén frissíteni a helyi változatot.

2 itthon.hu.zone

Ezután következnek az egyes zónákhoz tartozó zónafájlok. Elsőként az itthon.hu domain neveinek IP címmé feloldását lehetővé tevő zóna álljon itt:

/var/named/namedb/itthon.hu.zone

```
@           IN    SOA     tuzfalam.itthon.hu.  root.itthon.hu. (
                                  2003110702  ; serial
                                  10800       ; Refresh
                                  3600        ; Retry
                                  604800      ; Expire
                                  86400 )     ; Minimum

            IN    NS        tuzfalam.itthon.hu.


tuzfalam    IN    A         192.168.0.1
sinister    IN    A         192.168.0.10
niki        IN    A         192.168.0.11
laptop      IN    A         192.168.0.12

; Aliases:

babka       IN CNAME        niki
xsak        IN CNAME        sinister
bsd         IN CNAME        tuzfalam
gw          IN CNAME        tuzfalam
ns          IN CNAME        tuzfalam

;############################################################
;# Other machines....
gep100  IN A  192.168.0.100
gep101  IN A  192.168.0.101
gep102  IN A  192.168.0.102
gep103  IN A  192.168.0.103
gep104  IN A  192.168.0.104
gep105  IN A  192.168.0.105
gep106  IN A  192.168.0.106
gep107  IN A  192.168.0.107
```

```
gep108  IN A  192.168.0.108
... és a többi ... és a többi ...
gep243  IN A  192.168.0.243
gep244  IN A  192.168.0.244
gep245  IN A  192.168.0.245
gep246  IN A  192.168.0.246
gep247  IN A  192.168.0.247
gep248  IN A  192.168.0.248
gep249  IN A  192.168.0.249
gep250  IN A  192.168.0.250
```

A kukaccal kezdődő sorban sok fontos beállítást eszközölünk. A „tuzfalam.itthon.hu.” megadja a szervergép nevét. Fontos itt a név után álló pont! Ezután egy „adminisztrációs” e-mail címet adhatunk meg, de a kukacot ki kell cserélni pontra.

A Serial sorában egy egyedi sorszámot kell megadni, amit növelni szabad csak, s ebből fogja tudni induláskor a named, hogy változás történt a zónában, ezért minden módosításkor növelni kell az értékét. Célszerű ide a módosítás dátumát és idejét megadni, de lehet más is. A Refresh sorában másodpercben adja meg azt a időközt, amilyen gyakran az esetlegesen létező slave névszerverek frissítik adatbázisukat (10800 s = 3 óra). Mi csak egy master névszervert állítunk be. A Retry értékeként az esetleges sikertelen lekérések közti újrapróbálkozások intervallumát adjuk meg (3600 s = 1 óra). Az Expire értéke meghatározza a másodlagos szerverek adatainak kifutási (érvénytelenné válásának) idejét másodpercben (604800 s = 7 nap). A Minimum érték FIXME.

A következő szakaszban (NS sor) megadjuk, hogy a „tuzfalam.itthon.hu” gép lesz a névkiszolgáló.

Az „A”-s sorokban megadunk néhány kiemelt gépet a hálózatunkon, tulajdonképpen ezek a sorok adják az „adatbázist”. A tartomány nevét nem kell megadni.

Az „Aliases” megjegyzés utáni CNAME (canonical name) részben alias-okat, másodneveket adhatunk gépeinknek. Pl. A „tuzfalam” gépet elérhetem majd „bsd”, „gw”, „ns” neveket is...

Ezek után már csak töltelékként az egyéb, hálózatunkra ideiglenesen csatlakozó gépeknek adunk lehetőséget névvel létezni. A következő DHCP-t leíró rész szerint is minden „ki nem emelt” gép ilyen, gepFIXME.itthon.hu névvel lesz elérhető.

3 itthon.hu.rev

/var/named/namedb/itthon.hu.rev

```
@       IN   SOA    tuzfalam.itthon.hu.      root.itthon.hu. (
                          2003110702 ; Serial
                          10800      ; Refresh
                          3600       ; Retry
                          604800     ; Expire
                          86400 )    ; Minimum

        IN   NS     tuzfalam.itthon.hu.
1       IN   PTR    tuzfalam.itthon.hu.
10      IN   PTR    sinister.itthon.hu.
11      IN   PTR    niki.itthon.hu.

;############################################################
;# Other machines...

100             IN PTR gep100.itthon.hu.
101             IN PTR gep101.itthon.hu.
102             IN PTR gep102.itthon.hu.
```

```
103             IN PTR gep103.itthon.hu.
104             IN PTR gep104.itthon.hu.
105             IN PTR gep105.itthon.hu.
106             IN PTR gep106.itthon.hu.
107             IN PTR gep107.itthon.hu.
108             IN PTR gep108.itthon.hu.
... és a többi ... és a többi ...
243             IN PTR gep243.itthon.hu.
244             IN PTR gep244.itthon.hu.
245             IN PTR gep245.itthon.hu.
246             IN PTR gep246.itthon.hu.
247             IN PTR gep247.itthon.hu.
248             IN PTR gep248.itthon.hu.
249             IN PTR gep249.itthon.hu.
250             IN PTR gep250.itthon.hu.
```

E fájlnál nagyjából minden ugyanaz, mint az előzőnél, csak pepitában visszafelé. Magáért beszél (?), de a pontokat ne felejtsük az itthon.hu végéről...

4 localhost.zone

/var/named/namedb/localhost.zone

```
localhost.   IN  SOA   tuzfalam.itthon.hu. xsak.c2.hu. (
               2003110701   ; Serial
               10800        ; Refresh (3 hour)
               3600         ; Retry (1 hour)
               604800       ; Expire (1 week)
               86400 )      ; Minimum TTL (1 day)

localhost.   IN NS     tuzfalam.itthon.hu.
             ;IN NS     ns2.itthon.hu.

localhost.   IN A      127.0.0.1
```

Ez a fájl azért fontos, hogy helyesen oldhassuk fel a „localhost” nevet.

5 localhost.rev

/var/named/namedb/localhost.rev

```
0.0.127.in-addr.arpa. IN  SOA  tuzfalam.itthon.hu.  xsak.c2.hu. (
                      2003110701   ; Serial
                      10800        ; Refresh (3 hours)
                      3600         ; Retry (1 hour)
                      604800       ; Expire (1 week)
                      86400 )      ; Minimum TTL (1 day)
; Name Server (NS) records.
0.0.127.in-addr.arpa.   IN  NS   tuzfalam.itthon.hu.
;                        IN  NS   ns2.itthon.hu.

; Only one PTR record:
1.0.0.127.in-addr.arpa. IN  PTR  localhost.
```

Itt mondjuk meg, hogy a 127.0.0.1 IP feloldása bizony a „localhost” név.

6 all-zero.rev

/var/named/namedb/all-zero.rev

```
0.in-addr.arpa. IN  SOA tuzfalam.itthon.hu.  xsak.c2.hu. (
                20031100701  ; Serial
                10800        ; Refresh (3 hours)
                3600         ; Retry (1 hour)
                604800       ; Expire (1 week)
                86400        ; Minimum TTL (1 day)

0.in-addr.arpa.       IN  NS  tuzfalam.itthon.hu.
;                      IN  NS  ns2.itthon.hu.
```

Itt a csupa nullás IP címeket oldjuk fel.

7 all-one.rev

/var/named/namedb/all-one.rev

```
255.in-addr.arpa. IN SOA tuzfalam.itthon.hu. xsak.c2.hu. (
                20031100701  ; Serial
                10800        ; Refresh (3 hours)
                3600         ; Retry (1 hour)
                604800       ; Expire (1 week)
                86400        ; Minimum TTL (1 day)

255.in-addr.arpa.       IN  NS  tuzfalam.itthon.hu.
;                      IN  NS  ns2.itthon.hu.
```

Ennek segítségével tudhatjuk meg a 255.255.255.255 IP cím jelentését. FIXME

8 Névkiszolgáló kezelése

Az **rndc** segédprogrammal tudjuk leállítani a névkiszolgálót, vagy állapotáról információkat kérni.

Leállítás:

```
rndc stop
```

Állapotinformáció:

```
tuzfalam# rndc status
number of zones: 5
debug level: 0
xfers running: 0
xfers deferred: 0
soa queries in progress: 0
query logging is OFF
server is up and running
```

Indítás:

```
named
```

Lekérdezés (2 példa):

```
tuzfalam# host tuzfalam
tuzfalam.itthon.hu has address 192.168.0.1
tuzfalam# host sinister
tuzfalam.itthon.hu has address 192.168.0.10
```

A névkiszolgálót, ha a rendszer indulásakor szeretnénk elindítani módosítsuk a **/etc/rc.conf** fájlban a named_flags=”NO” sort erre:

```
named_flags=””
```

FIXME? Nálam ez a módszer nem jött be. Nem tudom miért, de ha boot-kor indítom a named-et, akkor az megall (logokban nincs nyoma), ezért inkább az ADSL kapcsolat elindultakor, a /etc/ppp/ppp.linkup fájl segítségével indítom. Lásd a „Szolgáltatások kezelése a PPP kapcsolat felépülése/lebontása esetén” pontot.

5 DHCP beállítása

A belső hálózatunk részére adjunk automatikusan IP címet, s adjuk meg az összes adatot az internet eléréséhez DHCP-vel. A kis szervergépünk minden hálózatra csatlakozó gépnek fog adni egy IP címet, megmondja, hogy őt használjuk névszerverként és alapértelmezett átjáróként az internet felé.

Néhány gépnek mindig ugyanazt az IP-t szeretném adni, ezért őket „kiemelem”, s MAC címük alapján azonosítom. A MAC cím egy olyan hexadecimális számsor, amely minden egyes hálózati kártyánál egyedi, nincs belőle két egyforma.

/etc/dhcpd.conf

```
#     $OpenBSD: dhcpd.conf,v 1.1 1998/08/19 04:25:45 form Exp $
#
# DHCP server options.
# See dhcpd.conf(5) and dhcpd(8) for more information.
#

# Network:          192.168.1.0/255.255.255.0
# Domain name:            my.domain
# Name servers:           192.168.1.3 and 192.168.1.5
# Default router:   192.168.1.1
# Addresses:        192.168.1.32 - 192.168.1.127
#
shared-network LOCAL-NET {
option domain-name "itthon.hu";
option domain-name-servers tuzfalam.itthon.hu;
option subnet-mask 255.255.255.0;
option routers 192.168.0.1;

      subnet 192.168.0.0 netmask 255.255.255.0 {
      range 192.168.0.100 192.168.0.250;
      }

      host sinister {
      hardware ethernet 00:60:52:0B:4D:E7;
      fixed-address 192.168.0.10;
      option subnet-mask 255.255.255.0;
      }

      host tuzfalam {
      hardware ethernet 00:e0:29:18:7c:87;
      fixed-address 192.168.0.1;
```

```
        option subnet-mask 255.255.255.0;
        }

        host niki {
        hardware ethernet 00:e0:98:ad:d4:d9;
        #hardware ethernet 00:00:00:00:00:00;
        fixed-address 192.168.0.11;
        option subnet-mask 255.255.255.0;
        }

}
```

Az „option domain-name "itthon.hu";” sor megadja a használandó DNS tartománynevet, az „option domain-name-servers tuzfalam.itthon.hu;” megadja, hogy a tűzfalunkat használja a kliens névszerverként. Az „option subnet-mask 255.255.255.0;” az alhálózati maszkot tudatja a kliensekkel, míg a végén az „option routers 192.168.0.1;” alapértelmezett átjáróként szintén tűzfalunkat határozza meg.

A host sorok és az utána következő kapcsos-zárójeles szakaszban megadunk egy gépnevet, s az ahhoz tartozó MAC address-t, valamint azt, hogy milyen fix IP címet kapjon az adott gép. Aki nem ezen gépek közül kér címet, az a jelen példában a 192.168.0.100 - 192.168.0.250 címek között kap egyet sorban az elejétől kezdve.

Meg lehet mondani azt is, hogy melyik hálózati csatolón osszon címeket, ezt a /etc/dhcpd.interfaces fájlban tehetjük meg:

```
#       $OpenBSD: dhcpd.interfaces,v 1.1 1998/08/19 04:25:45 form Exp $
#
# List of network interfaces served by dhcpd(8).
#
# ep0
# ed0 le0
# de1
ne3
```

Ha a DHCP kiszolgálót a rendszer indulásakor kívánjuk indítani, akkor a **/etc/rc.conf** fájlban írjuk át a „dhcp_flags=”NO”” sort írjuk át valahogy így:

```
dhcpd_flags=”-q”
```

Én ezt a démont sem indítom rendszer indulásakor, hanem inkább az ADSL kapcsolat létrejöttekor a /etc/ppp/ppp.linkup fájl segítségével. Lásd a „Szolgáltatások kezelése a PPP kapcsolat felépülése/lebontása esetén” pontot.

6 Szolgáltatások kezelése a PPP kapcsolat felépülése/lebontásakor

Az internethez köthető szolgáltatásokat célszerű az internet kapcsolattal együtt ki illetve bekapcsolni. Példánk ADSL-ről szól, ezért itt a korábban már említett fájlok szerkesztéséről van szó:

- /etc/ppp/ppp.linkup
- /etc/ppp/ppp.linkdown

Amikor az ADSL kapcsolat létrejön, lefut a ppp.linkup, mikor pedig lebomlik, akkor a ppp.linkdown. E fájlokat az alább látható módon kell megszerkeszteni. Figyeljünk oda, mert a fájl formátuma kötött, a sorok előtti szóköznek is szerepelnie kell.

/etc/ppp/ppp.linkup

```
MYADDR:
 ! sh -c "/sbin/ifconfig pflog0 up"
 ! sh -c "/sbin/pfctl -F all"
 ! sh -c "/sbin/pfctl -R -f /etc/pf.conf"
 ! sh -c "/sbin/pfctl -N -f /etc/nat.conf"
 ! sh -c "/sbin/pfctl -e"
 ! sh -c "/usr/sbin/rndc stop ; /usr/sbin/named"
 ! sh -c "/sbin/pflogd"
# ! sh -c "/usr/local/ntpd -p /var/run/ntpd.conf"
 ! sh -c "cp /etc/resolv.conf-working /etc/resolv.conf"
 ! sh -c "/usr/sbin/dhcpd"
 ! sh -c "/usr/local/sbin/ddclient -daemon=0 -syslog -verbose -noquiet"
 ! sh -c "/usr/local/sbin/sendmyip.sh"
# !bg sh -c "/usr/local/sbin/do_ipcheck"
```

Nézzük soronként:

```
MYADDR:
 ! sh -c "/sbin/ifconfig pflog0 up"
```

Az első sor elindítja a **pflog0** virtuális interfészt, ami a pf csomagszűrő naplózásához ad egy csatolófelületet.

```
 ! sh -c "/sbin/pfctl -F all"
```

Ezután a biztonság kedvéért kitöröljük (F=flush) az összes pf szabályt.

```
 ! sh -c "/sbin/pfctl -R -f /etc/pf.conf"
```

Betöltjük a szűrési szabályokat.

```
 ! sh -c "/sbin/pfctl -N -f /etc/nat.conf"
```

Betöltjük a NAT (címfordítási) szabályokat

```
 ! sh -c "/sbin/pfctl -e"
```

Aktiváljuk a csomagszűrést (e=enable).

```
 ! sh -c "/usr/sbin/rndc stop ; /usr/sbin/named"
```

A DNS kiszolgálót újraindítjuk. Ez szükséges a megváltozott hálózati környezet (tun0 csatoló létrejön, más lesz a DNS-kiszolgáló, az alapértelmezett átjáró...)

```
 ! sh -c "/sbin/pflogd"
```

Ezzel elindul a pf naplózó démonja. Ez később hibakeresésnél hasznos lehet.

```
# ! sh -c "/usr/local/ntpd -p /var/run/ntpd.conf"
```

Ha akarnánk, lehetne egy belső időkiszolgálót is indítani. Én ezt most nem...

```
 ! sh -c "cp /etc/resolv.conf-working /etc/resolv.conf"
```

Amikor az ADSL (pontosabban a ppp) kapcsolat felépül, a ppp démon átírja a /etc/resolv.conf fájlt úgy, hogy az ADSL kliensként kapott DNS szervereket használja. Nekünk azt kell megoldanunk, hogy a gép a névfeloldásra önmagát használja. Tehát ennek a fájlnak léteznie kell, s tartalma legyen ilyesmi:

```
nameserver 192.168.0.1
lookup file bind
search itthon.hu
```

```
! sh -c "/usr/sbin/dhcpd"
```

Elindítjuk a jól beállított dhcp kiszolgálónkat.

```
! sh -c "/usr/local/sbin/ddclient -daemon=0 -syslog -verbose -noquiet"
```

A dinamikus IP címünket állítjuk be ezzel a szkripttel. Használatához persze először regisztrálni kell egy fiókot pl. a DynDNS honlapján, majd azt be kell állítani a későbbiekben leírtak (s a józan eszünk) szerint.

```
! sh -c "/usr/local/sbin/sendmyip.sh"
```

Írtam egy kis szkriptet, ami elküldi egy általam megadott (webes e-mail) címre az aktuális IP címemet. Ez akkor kellhet, ha a DynDNS becsődöl. Megnézem az utolsó levelet az interneten, s oda csatlakozom ssh-val. A szkript így fest:

```
#! /bin/sh
#
datum=`date +%Y%m%d-%H%M%S`
mailcim="xsak@c2.hu"
IP=$(/sbin/ifconfig tun0 | awk '/netmask/{print $2}')
echo "A jelenlegi ip-cim: $IP " > /tmp/tmp.$datum
sleep 1
host $IP >> /tmp/tmp.$datum
echo "_____" >> /
tmp/tmp.$datum
echo $datum >> /tmp/tmp.$datum
echo "_____" >> /
tmp/tmp.$datum
ifconfig -a >> /tmp/tmp.$datum
echo "_____" >> /
tmp/tmp.$datum
cat /tmp/tmp.$datum | /usr/bin/mail -s "$datum" $mailcim
rm -f /tmp/tmp.$datum
```

Ahhoz, hogy a fenti szkript ténylegesen elküldje a levelet, a levelező szerverünket (sendmail, ami alapból benne van az OpenBSD-ben) be kell állítani. Ezt én egy nagyon nem elegáns módon oldottam meg FIXME. A sendmailt egy sendmail.mc fájlon keresztül illik konfigurálni, s abból egy makróval a sendmail.cf konfigurációs fájlt létrehozni. Ehelyett a sendmail.cf-et most közvetlenül átszerkesztjük, amit mindenhol főbenjáró bűnként emlegetnek, s ERŐSEN ELLENJAVALT! Bár azt is mondják, hogy aki egyszer sem szerkesztett még sendmail.cf-et, annak nincs szíve, de aki egynél többször, annak nincs esze :-).

Szóval két sort kell átszerkeszteni. Az egyik a „Cw”-vel kezdődő, ahova a tuzfal gép hosztnevét adjuk meg, a másik pedig a „DS”-el kezdődő, ahova a smarthostként (továbbító levelezőszerver) üzemelő gépet adjuk meg.

/etc/mail/sendmail.cf

```
Cwtuzfalam
```

…

```
Dsmail.szolgaltatom.hu
```

Természetesen ez utóbbinál mindenki a saját szolgáltatója által megadott smtp

kiszolgálót írja be. A sendmail következő újraindulása (+ /etc/rc.conf sendmail-re vonatkozó beállításai) után elvileg a mail parancssoros program tud levelet küldeni a világba (de fogadni így még nem).

Végül az utolsó, kikommentezett sor:

```
# !bg sh -c "/usr/local/sbin/do_ipcheck"
```

Ez is egy saját szkript, hagyjuk...

/etc/ppp/ppp.linkdown

```
MYADDR:
 ! sh -c "/sbin/pfctl -F all -d"
 ! sh -c "/usr/sbin/rndc stop ; /usr/sbin/named"
# ! sh -c "cp /etc/resolv.conf-noinet /etc/resolv.conf"
 ! sh -c "kill `cat /var/run/pflogd.pid`"
```

Mit is csinálunk itt?

```
MYADDR:
 ! sh -c "/sbin/pfctl -F all -d"
```

Ezen sor kiüríti az összes és NAT szabályt, illetve inaktívvá teszi a csomagszűrést (d=disable).

```
 ! sh -c "/usr/sbin/rndc stop ; /usr/sbin/named"
```

Újraindítja a névszervert.

```
# ! sh -c "cp /etc/resolv.conf-noinet /etc/resolv.conf"
```

Ez arra szolgált, hogy egy érvényes /etc/resov.conf fájlt adjon (ami a névfeloldást határozza meg) akkor is, ha nincs internet kapcsolat. Ezt most nem használjuk.

```
 ! sh -c "kill `cat /var/run/pflogd.pid`"
```

Leállítja a pflogd-t, a csomagszűrő naplózódémonját.

Ezzel remélhetőleg rendezni tudjuk a kapcsolat felépülésekor történő eseményeket.

7 Dinamikus DNS beállítása

Az én ADSL előfizetésem úgy működik, hogy szolgáltatóm 24 óránként megszakítja a vonalat. Szerencsére az OpenBSD ppp-je úgy van beállítva, hogy az megpróbál újra csatlakozni. Minden (újra)csatlakozáskor másik IP-címet kapunk, ami azt okozza, hogy legalább naponta más-más néven lehetne gépünket elérni.

Vannak olyan szolgáltatók az interneten, akiknél regisztrálhatunk (ingyen) egy olyan aldomaint, amivel mindig ugynazzal a névvel hivatkozhatunk az aktuális IP címre. Ez azzal jár (általában), hogy egy kliensprogramot kell a gépünkre telepíteni, ami minden csatlakozás után bejelentkezik a dinamikus DNS szolgáltatóhoz, és közli az új, megváltozott IP címet. Ezután már az új IP-re mutat az „állandó” nevünk.

Először tehát szerezzünk be egy azonosítót, s hozzá egy klienst. Én a DynDNS szolgáltatását használom, ami elérhető itt: http://www.dyndns.org/. Válasszunk kliensprogramot is, az oldalon sok lehetőség szerepel. Jelen helyen a **ddclient** beállítását írom le (ez egy perl szkript).

A ddclientet telepíthetjük a ports fából is. A ports egy olyan könyvtárrendszer, amiben bizonyos OpenBSD-re portolt (átírt) programok lefordításához szükséges fájlokat tartalmaz. Egy ports telepítésekor letölti a forrást, lefordítja, majd telepíti azt. Függőségeket is kezeli, tehát ha a telepíteni kívánt csomag függ egy másiktól, akkor azt is letölti és telepíti. A ports fát feltehetjük az OpenBSD ftp-ről letöltött **ports.tar.gz** fájl kicsomagolásával a /usr/ alá. Ha a CD készítésénél pont a leírás szerint jártunk el, akkor a CD-n rajta van a ports is.

Tehát a telepítést végezzük úgy, hogy belépünk a **/usr/ports/net/ddclient** könyvtárba, s ott kiadjuk a make install parancsot (élő internetkapcsolat mellett).

```
cd /usr/ports/net/ddclient
make install
tuzfalam# cd /usr/ports/net/ddclient
tufalam# make install
===>  Checking files for ddclient-3.6.3
>> ddclient-3.6.3.tar.gz doesn't seem to exist on this system.
cp: /cdrom/distfiles//ddclient-3.6.3.tar.gz: No such file or directory
>>  Attempting  to  fetch  /usr/ports/distfiles/ddclient-3.6.3.tar.gz  from
http://members.rogers.com/ddclient/pub/.

                                                    0%                      |
|     0       --:-- ETA
                                   94%                                      |
*****************************************************************************
*     | 32768       00:00 ETA
100%                                                                        |
*****************************************************************************
******| 34497       00:01
>> Checksum OK for ddclient-3.6.3.tar.gz. (sha1)
===>  Extracting for ddclient-3.6.3
===>  Patching for ddclient-3.6.3
===>  Configuring for ddclient-3.6.3
===>  Faking installation for ddclient-3.6.3
install  -c  -o  root  -g  bin  -m  555  /usr/ports/net/ddclient/w-ddclient-
3.6.3/ddclient-3.6.3/ddclient             /usr/ports/net/ddclient/w-ddclient-
3.6.3/fake-i386/usr/local/sbin/
install  -d  -o  root  -g  bin  -m  755  /usr/ports/net/ddclient/w-ddclient-
3.6.3/fake-i386/usr/local/share/examples/ddclient
install  -d  -o  root  -g  bin  -m  755  /usr/ports/net/ddclient/w-ddclient-
3.6.3/fake-i386/usr/local/share/doc/ddclient
install  -c  -o  root  -g  bin  -m  444  /usr/ports/net/ddclient/w-ddclient-
3.6.3/ddclient-3.6.3/sample*   /usr/ports/net/ddclient/w-ddclient-3.6.3/fake-
i386/usr/local/share/examples/ddclient
install  -c  -o  root  -g  bin  -m  444  /usr/ports/net/ddclient/w-ddclient-
3.6.3/ddclient-3.6.3/README*   /usr/ports/net/ddclient/w-ddclient-3.6.3/fake-
i386/usr/local/share/doc/ddclient
===>  Building package for ddclient-3.6.3
Creating package /usr/ports/packages/i386/all/ddclient-3.6.3.tgz
Creating gzip'd tar ball in '/usr/ports/packages/i386/all/ddclient-3.6.3.tgz'
Link to /usr/ports/packages/i386/ftp/ddclient-3.6.3.tgz
Link to /usr/ports/packages/i386/cdrom/ddclient-3.6.3.tgz
===>   Installing  ddclient-3.6.3  from  /usr/ports/packages/i386/all/ddclient-
3.6.3.tgz
Adding /usr/ports/packages/i386/all/ddclient-3.6.3.tgz

tuzfalam#
```

Ha kézzel telepítjük, a ddclient letöltése után azt kicsomagoljuk egy ideiglenes könyvtárba, majd onnan a végrehajtható fájlt (ddclient) áthelyezzük egy útvonalba szereplő könyvtárba (vagy bárhova, csak később tudjuk, hova tettük).

Ezután beállíthatjuk a klienst. Nálam a /etc/ddclient.conf fájl így néz ki:

/etc/ddclient.conf

```
syslog=yes
use=if, if=tun0
protocol=dyndns2
login=xsak
password=NagyonTitkosJelszavam

server=members.dyndns.org
protocol=dyndns2
xsak.azendinamikusdnsdomainem.org
```

A fájlt persze ezután a root tulajdonába kell helyezni 600 jogokkal. A fájlban módosítandó a felhasználónév és a jelszó, meg persze a végén a regisztrált domain-név.

Ezután a DynDNS cím beállítása történhet a parancssorból egy hasonló parancssal:

```
/usr/local/sbin/ddclient -daemon=0 -syslog -verbose -noquiet
```

Az előző részben láthattuk, hogy a /etc/ppp/ppp.linkup fájlban szereplő bejegyzés szerint induláskor frissíti a név-cím összerendelést. Azért én írtam egy nagyon egyszerű szkriptet, ami negyedóránként cron-ból futtatva megnézi, hogy egyezik-e a DynDNS általi névhez tartozó IP cím a mi címünkhöz. Ha nem, akkor megpróbálja beállítani. Íme:

/usr/local/sbin/chkdyndns.sh

```
#! /bin/sh
echo -n " DynDnS Vs. CurrentIP (`date \"+%Y.%m.%d-%H%M%S\"`) ...  "
ipnow=`/sbin/ifconfig tun0 | awk '/netmask/{print $2}'`
xsip=`host xsak.azendinamikusdomainem.org | awk '{ print $4 }'`
if [ $ipnow != $xsip ]; then
        echo " "
        echo "Different! Trying to set again..."
                /usr/local/sbin/ddclient  -daemon=0  -syslog  -verbose  >  /
var/log/ddclient.last 2>&1
else
        echo "Same."
fi
```

A „**crontab -e**” parancs kiadásával így helyeztem el a root crontab-jában:

```
# Setting up xsak.azendinamikusdomainem.org at dyndns.org...
*/15    *       *       *       *       /usr/local/sbin/chkdyndns.sh >> /
var/log/chkdyndns.log 2>&1
```

A művelet negyedóránként fut, s eredménye megtekinthető a /var/log/chkdyndns.log fájlban.

8 Címfordítás és tűzfal beállítása

Most jön a legnehezebb és legkényesebb rész, a tűzfalazás beállítása. FIXME Jó lenne, ha egy szakértő ezt véleményezné!!! FIXME Az OpenBSD csomagszűrője a pf — készítőinek állítása szerint — egy nagyon jó, biztonságos és stabil program. Mindenképpen tanulmányozzuk át a http://www.openbsd.org/faq/pf/index.html címen található leírást, nagyon hasznos! A csomagszűrőt a pfctl programon keresztül állítgathatjuk, s szövegfájlokba írjuk a tűzfal- és címfordítási szabályokat.

A pf akkor indul el a gép indulásakor, ha a **/etc/rc.conf** fájlban a következőhöz

hasonló bejegyzéseket tesszük:

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

De mi nem a gép indulásakor kezdjük a csomagszűrést, hanem a már leírt ppp.linkup fájlban, a ppp (az ADSL kapcsolat) indulásakor.

Fontos bekapcsolnunk az IP továbbítást a belső hálózat és az ADSL vonal között. Ehhez a /etc/sysctl.conf fájlban lennie kell egy ilyesmi sornak (egészen pontosan csak ki kell venni a # jelet a sor elejéről):

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

Készítünk majd egy **/etc/nat.conf** és egy **/etc/pf.conf** fájlt, amit „betöltünk" a csomagszűrőbe, ezzel biztosítva az átirányításokat és szűréseket. A **/etc/nat.conf** fájlba kerülnek a címfordítási, s értelemszerűen a **/etc/pf.conf** fájlba a csomagszűrési szabályok.

1 Címfordítás (NAT)

Nézzük előbb a címfordítást:

/etc/nat.conf

```
Ext = tun0
nat on $Ext from 192.168.0.0/24 to any -> $Ext
rdr on ne3 proto tcp from any to any port 21 -> 127.0.0.1 port 8081
rdr on $Ext proto tcp from any to any port 5865 -> 192.168.0.10 port 22
# rdr on $Ext proto tcp from any to any port 6881:6889 -> 192.168.0.10 port
6881:*

# More anchored redirects to come...
rdr-anchor redirect
```

Elég egyszerűnek tűnik, de lássuk mit jelent részleteiben.

```
Ext = tun0
```

Makróban definiáljuk a külső interfészt a könnyű változtathatóság miatt. Ez ADSL esetén a már említett tun0, egy virtuális csatoló.

```
nat on $Ext from 192.168.0.0/24 to any -> $Ext
```

Ez a sor veszi rá a pf-et a NAT (címfordítás) elvégzésére az $Ext csatolón A belső hálózati tartomány részére. A belső címekről induló csomagokat a külső csatolóra fordítja.

```
rdr on ne3 proto tcp from any to any port 21 -> 127.0.0.1 port 8081
```

Ebben a sorban az ftp elérést biztosítjuk, ami tűzalaknál általában problémás. Jelen esetben minden kifelé s a 21-es portra irányuló kérést elkap a tűzfal és önmagára, önmaga 8081-es portjára irányít. A belső hálózaton ezen a porton egy **ftp-proxy** nevű program figyel, ami az ftp elérést biztosítja. Az rdr kifejezés redirect-et jelent, s itt meg kell adnunk, hogy melyik hálózati csatolón történjen az átirányítás (mi a belsőt adjuk meg!). Az ftp elérés beállítása még nincs kész! A **/etc/inetd.conf** fájlban legyen egy ilyen sor:

```
127.0.0.1:8081 stream tcp nowait root /usr/libexec/ftp-proxy ftp-
proxy -n -u proxy -m 55000 -M 57000 -t 180
```

Ezzel elérjük, hogy az inetd daemon figyeljen a sajátgép 8081-es portján, s az azon kéréseket átadja az ftp-proxy programnak. Az ftp-proxy kapcsolói a következőket jelentik:

-n	Aktiválja a NAT módot, ami a passzív elérés miatt fontos
-u proxy	a „proxy” felhasználó nevében fut
-m 55000	Adatkapcsolat minimális portszáma
-M 57000	Adatkapcsolat maximális portszáma
-t 180	Időtúllépési korlát

Persze az inetd-t is indítani kell, legyen ez a gép indulásakor. Írjuk (át) a /etc/rc.conf fájlba a következőhöz hasonlót:

```
inetd=YES
```

De folytassuk tovább a nat.conf taglalását:

```
rdr on $Ext proto tcp from any to any port 5851 -> 192.168.0.10 port 22
```

Ez egy példa arra, hogyan lehet egy belső számítógépet elérni ssh-n keresztül. A sor jelentése: az összes 5851-es portra érkező kérést továbbítsd a 192.168.0.10-es IP című gép 22-es (SSH) portjára.

```
# rdr on $Ext proto tcp from any to any port 6881:6889 -> 192.168.0.10 port
6881:*
```

Ez egy példa arra, hogy mit kell beélesíteni (kivenni a megjegyzés-jelet) ahhoz a NAT részen, hogy a Bittorrent rendszert használni lehessen. Ezt én nem hagyom alapértelmezetten, mert kézzel, anchor-ral megoldva illesztem be e szabályt amikor az szükséges (lásd a **Bittorrent beállítása** pontot).

```

# More anchored redirects to come...
rdr-anchor redirect
```

Ez egy horgony. A horgonyok segítségével lehet később, működés közben szabályokat beilleszteni a meglévő szabályrendszerbe. Ez jelenti azt a pontot, ahova be lehet illeszteni később az erre a horgonyra vonatkozó szabályokat. Miért is jó ez? Mert utólag bármikor ki lehet venni a szabályt, illetve csak akkor tesszük be, amikor szükséges (például egy port megnyitása).

2 Csomagszűrő

A /etc/pf.conf fájlba írt szabályok meghatározóak, mindig gondoljuk át, mit teszünk bele. Ez a fájl egy interneten talált szabályrendszeren alapul, azt módosítottam saját igényeim és tudásom szerint. Később részletesen bemutatom, melyik sor, mire szolgál...

/etc/pf.conf

```
#-----------------------------------------------------------------------
# PF ruleset, 11 dec. 2001
#
# Liberally adapted from the pf man page, the OpenBSD "Network How-To",
# and my own rulesets.
#-----------------------------------------------------------------------

#-----------------------------------------------------------------------
# Definitions
Ext = "tun0"             # External interface
Int = "ne3"              # Internal interface
Loop = "lo0"             # Loopback interface
IntNet="192.168.0.0/24" # Internal network
```

```
NoRoute   =   "{   127.0.0.1/8,   192.168.0.0/16,   172.16.0.0/12,   10.0.0.0/8,
255.255.255.255/32 }"

InServicesTCP = "{ ssh, smtp, auth, http, https, pop3 }"
#InServicesTCP = "{ ssh auth }"
#InServicesUDP = "{ domain }"
# port 1863 = MSN
# port 5190 = ICQ
OutServicesTCP = "{ http, https, smtp, pop3, imap, whois, domain, ssh, telnet,
ftp, ftp-data, nntp, auth, rsync, 1863, 5190, 8880 }"
OutServicesUDP = "{ ntp, domain }"

XMMS = "{ 6000, 7500, 8000, 8004, 8044, 8034, 8052, 8038, 8010, 8400, 8014,
8026, 8048, \
          8002, 8024, 8028, 8080 }"
RealAudio = "{ 554, 7070, 8080 }"

IRCports = "{ 6667, 6666, 6668, 6669 }"
#               irc.hu         papucs.vagyok.hu irc.sote.hu     extra.irc.hu
IRCservers    =    "{    157.181.1.129,    192.188.242.121,    193.224.51.150,
195.70.37.253}"
FreeNodeIRCServers    =    "{    128.193.0.29    128.193.0.47    130.239.18.172
208.185.243.68 213.28.116.205 82.96.64.2 }"
#  FreeNodeIRCServers  =  "{  irc.freenode.net  }"  #  Nem  jo,  mert  amig  nincs
nevfeloldas, nem tud a pfctl -R befejezodni...

#CVSup    hosts:                   cvsup.hu.freebsd.org    cvsup.uk.openbsd.org
cvsup.hu.openbsd.org
CVSupServers = "{ 193.225.13.161, 194.242.157.43, 152.66.243.8 }"
CVSupPorts = "{ 5999 }"

JabberPorts = "{ 5222, 5223 }"

DynDNSServer = "{ 63.208.196.94 }"
DynDNSPorts = "{ 8245 }"

FreeDBhosts = "{ 64.71.163.204, 130.179.31.49, 193.166.235.14, 193.201.200.74,
195.37.77.133, 203.16.234.30 }"
FreeDBports = 888

#--------------------------------------------------------------------------
#--------------------------------------------------------------------------
# Clean up fragmented and abnormal packets
# By default in pf, packets which contain IP options are blocked. Good.
scrub in on { $Ext, $Int } all
#--------------------------------------------------------------------------


#--------------------------------------------------------------------------
# ALTQ ADSL bandwidth resolution (http://www.benzedrine.cx/ackpri.html)
altq on $Ext priq bandwidth 100Kb queue { q_pri, q_def }
queue q_pri priority 7
queue q_def priority 1 priq(default)

pass  out  on  $Ext  proto  tcp  from  $Ext  to  any  flags  S/SA  keep  state  queue
(q_def, q_pri)
pass  in   on  $Ext  proto  tcp  from  any  to  $Ext  flags  S/SA  keep  state  queue
(q_def, q_pri)
#--------------------------------------------------------------------------


#--------------------------------------------------------------------------
```

```
# Defaults
# block and log everything
block             out log on $Ext           all
block             in  log on $Ext           all
block return-rst  out log on $Ext proto tcp all
block return-rst  in  log on $Ext proto tcp all
block return-icmp out log on $Ext proto udp all
block return-icmp in  log on $Ext proto udp all

block in  quick inet6 all
block out quick inet6 all
#--------------------------------------------------------------------------

#--------------------------------------------------------------------------
# loopback packets left unmolested
pass in quick on $Loop all
pass out quick on $Loop all
#--------------------------------------------------------------------------

#--------------------------------------------------------------------------
# Immediate blocks
# fuzz any 'nmap' attempt
block in log quick on $Ext inet proto tcp from any to any flags FUP/FUP
block in log quick on $Ext inet proto tcp from any to any flags SF/SFRA
block in log quick on $Ext inet proto tcp from any to any flags /SFRA

# don't allow anyone to spoof non-routeable addresses
block in log quick on $Ext from $NoRoute to any
block out log quick on $Ext from any to $NoRoute

# silently drop broadcasts (cable modem noise)
block in quick on $Ext from any to 255.255.255.255
#--------------------------------------------------------------------------



#--------------------------------------------------------------------------
# PASS rules

# ALL  -- we don't normally do that. For debugging only.
#pass out quick on $Ext all keep state

# pass in data mode connections for ftp-proxy running on this host.
pass in quick on $Ext inet proto tcp from any to any port > 49151 flags S/SA
keep state

# ICMP
pass out     quick on $Ext inet proto icmp all icmp-type 8 code 0 keep state
pass in  log quick on $Ext inet proto icmp all icmp-type 8 code 0 keep state

# Services we provide to the outside world
#pass in quick on $Ext inet proto udp from any to any port $InServicesUDP keep
state
pass in quick on $Ext inet proto tcp from any to any port $InServicesTCP flags
S/SA keep state

# Standard services we want to access in the world
pass out quick on $Ext inet proto udp from any to any port $OutServicesUDP
keep state
pass out quick on $Ext inet proto tcp from any to any port $OutServicesTCP
flags S/SA modulate state
```

```
# Special services
pass out quick on $Ext inet proto tcp from any to any port $XMMS flags S/SA
modulate state
pass out quick on $Ext inet proto tcp from any to any port $RealAudio flags
S/SA modulate state

# CVSup to cvsup.hu.freebsd.org
pass out quick on $Ext inet proto tcp from any to $CVSupServers port
$CVSupPorts flags S/SA modulate state

# Time server we use: time.kfki.hu (148.6.0.1)
pass out quick on $Ext inet proto tcp from any to 148.6.0.1 port time flags
S/SA modulate state

# Hungarian IRC servers:
pass out quick on $Ext inet proto tcp from any to $IRCservers port $IRCports
flags S/SA modulate state
# FreeNode IRC servers:
pass out quick on $Ext inet proto tcp from any to $FreeNodeIRCServers port
$IRCports flags S/SA modulate state

# FreeDB access:
pass out quick on $Ext inet proto tcp from any to $FreeDBhosts port
$FreeDBports flags S/SA modulate state

# Jabber communication:
pass out quick on $Ext inet proto tcp from any to any port $JabberPorts flags
S/SA modulate state

# DynDNS.org update
pass out quick on $Ext inet proto tcp from any to $DynDNSServer port
$DynDNSPorts flags S/SA modulate state

# BitTorrent connections
# pass out quick on $Ext inet proto tcp from any to any port { 6880><6889,
6969 } flags S/SAFR keep state
# pass in quick on $Ext inet proto tcp from any to any port 6880><6889 flags
S/SAFR keep state

# Other pass rules with anchor...
anchor passin
```

Ez így elsőre vadnak tűnhet, lássuk mit jelentenek az egyes sorok:

```
# Definitions
Ext = "tun0"             # External interface
Int = "ne3"              # Internal interface
Loop = "lo0"             # Loopback interface
IntNet="192.168.0.0/24" # Internal network
```

Néhány definíció. Itt ún. makrókat adunk meg, amikre egy névvel hivakozhatunk a későbbiekben a szabályrendszerben. Célszerű ilyeneket használni, hisz ha valami változik a konfiguráción, akkor azt csak egy helyen kell átvezetni. Megadjuk a külső interfész nevét (tun0 = ADSL csatolófelülete), a belső hálózat felé néző hálózati kártyát (ne3) és a loopback, azaz a visszacsatolás csatolót. Megadjuk még a belső hálózat IP cím tartományát is. Ezekre rendre a $Ext $Int $Loop és $IntNet névvel hivakozhatunk.

```
NoRoute = "{ 127.0.0.1/8, 192.168.0.0/16, 172.16.0.0/12, 10.0.0.0/8,
255.255.255.255/32 }"
```

Ide gyűjtöttük a nem route-olható címeket.

```
InServicesTCP = "{ ssh, smtp, auth, http, https, pop3 }"
#InServicesTCP = "{ ssh auth }"
#InServicesUDP = "{ domain }"
```

Portokra is hivatkozhatunk névvel (makróval). Ide gyűjtöttem azon portokat, amiket kintről érhetnek el, ezek kintről engedélyezve vannak. A listát lehet bővíteni, szűkíteni, de csak gondos odafigyeléssel. Mint látható, a portokat nem csak számmal, hanem a **/etc/services** fájlban szereplő nevével is jelölhetjük. Csak annyi portot adjunk meg, amennyit nagyon muszáj!

```
# port 1863 = MSN
# port 5190 = ICQ

OutServicesTCP = "{ http, https, smtp, pop3, imap, whois, domain, ssh, telnet,
ftp, ftp-data, nntp, auth, rsync, 1863, 5190, 8880 }"
OutServicesUDP = "{ ntp, domain }"
```

Itt soroltam fel a kintről elérhető szolgáltatásokhoz tartozó portokat, külön a TCP és UDP portokat. A 1863 az MSN azonnali üzenetküldő portja, míg az 5190 az ICQ-é. Ezeket kivehetjük, ha nem használjuk, ne legyen itt.

```
XMMS = "{ 6000, 7500, 8000, 8004, 8044, 8034, 8052, 8038, 8010, 8400, 8014,
8026, 8048, \
          8002, 8024, 8028, 8080 }"
RealAudio = "{ 554, 7070, 8080 }"
```

No igen, újabb biztonságcsökkentő porthalmaz. Az internetes rádiók csatlakoznak általában ezeken a portokon, de persze mindig van olyan rádió, ami ebben a listában sincs benne. Itt is igaz: csak akkor használjuk, ha szükség van rá a hálózatunkon.

```
IRCports = "{ 6667, 6666, 6668, 6669 }"
#               irc.hu          papucs.vagyok.hu irc.sote.hu     extra.irc.hu
IRCservers    =    "{    157.181.1.129,    192.188.242.121,    193.224.51.150,
195.70.37.253}"
FreeNodeIRCServers    =    "{    128.193.0.29    128.193.0.47    130.239.18.172
208.185.243.68 213.28.116.205 82.96.64.2 }"
# FreeNodeIRCServers = "{ irc.freenode.net }" # Nem jo, mert amig nincs
nevfeloldas, nem tud a pfctl -R befejezodni...
```

Ha használod az IRC csevegési lehetőségét, akkor kell ezeket beállítani. Jelen példában a magyar irc szerverek és az irc.freenode.net IP címei szerepelnek. FIXME FIXME Először beírtam névvel az irc.freenode.net-et, de ez nem vált be, mert a következő csatlakozáskor sokat időzött a PF azon, hogy még névfeloldást nem tudott csinálni, de próbálta mégis betölteni az erre vonatkozó szabályokat is. Inkább kézzel kiszedtem az IP címeket, s bepötyögtem. Probléma akkor van, ha ezek változnak (márpedig változnak időnként).

```
#CVSup    hosts:                  cvsup.hu.freebsd.org    cvsup.uk.openbsd.org
cvsup.hu.openbsd.org
CVSupServers = "{ 193.225.13.161, 194.242.157.43, 152.66.243.8 }"
CVSupPorts = "{ 5999 }"
```

Ha a belső hálózaton van FreeBSD vagy OpenBSD operációs rendszerű gép, akkor ezek frisssítéséhez szükséges néhány IP címre egy portot megnyitni.

```
JabberPorts = "{ 5222, 5223 }"
```

Jabber, egy újabb azonnali üzenetküldő rendszer, illetve annak portja.

```
DynDNSServer = "{ 63.208.196.94 }"
DynDNSPorts = "{ 8245 }"
```

Ez a két adat szükséges ahhoz, hogy dinamikus DNS-ünket tudjuk frissíteni.

```
FreeDBhosts = "{ 64.71.163.204, 130.179.31.49, 193.166.235.14, 193.201.200.74,
195.37.77.133, 203.16.234.30 }"
FreeDBports = 888
```

Itt az online CD adatbázis adatai kerülnek felsorolásra.

Eddig voltak a meghatározások, a definíciók. Most következnek maguk a szabályok...

```
# Clean up fragmented and abnormal packets
# By default in pf, packets which contain IP options are blocked. Good.
scrub in on { $Ext, $Int } all
```

A scrub a csomagok „normalizálását” jelenti. Ezen kívül a töredezett csomagokat is összeilleszti, ami megakadályoz bizonyos fajta támadásokat. Ezt elvégezzük mind a belső, mind a külső hálózati csatolón.

```
# ALTQ ADSL bandwidth resolution (http://www.benzedrine.cx/ackpri.html)
altq on $Ext priq bandwidth 100Kb queue { q_pri, q_def }
queue q_pri priority 7
queue q_def priority 1 priq(default)
pass out on $Ext proto tcp from $Ext to any flags S/SA keep state queue
(q_def, q_pri)
pass in  on $Ext proto tcp from any to $Ext flags S/SA keep state queue
(q_def, q_pri)
```

Alapesetben előfordulhatna, hogy nagy feltöltés esetén eléggé lelassul egy esetleges párhuzamosan futó letöltés sebessége, aminek az az oka, hogy a feltöltés sávszélességigénye és az ADSL aszimmetrikussága miatt nem jut elég sávszélesség a letöltéshez tartozó az ún. ACK (visszaigazoló) csomagok elküldésére, amitől a letöltés másik oldalán lévő szerver azt hiszi, elvesznek a csomagok, s lassítja küldési sebességét. Ezek a szabályok megpróbálják normalizálni a le/feltöltést, persze csodát ne várjunk. Egy otthoni hálózatban nem bír nagy jelentőséggel ez, ha nincs rá szükségünk elhagyható.

```
# Defaults
# block and log everything
block             out log on $Ext           all
block             in  log on $Ext           all
block return-rst  out log on $Ext proto tcp all
block return-rst  in  log on $Ext proto tcp all
block return-icmp out log on $Ext proto udp all
block return-icmp in  log on $Ext proto udp all

block in  quick inet6 all
block out quick inet6 all
```

Minden jó tűzfal (s talán a mienk is idesorolható (egyszer majd)) azon az elven alapul, hogy alapértelmezésként mindent tiltunk, csak megengedünk bizonyos eléréseket. Ez a néhány sor letilt minden forgalmat a külső csatolónkon, s a kapcsolat adatait naplózza a pflog0 virtuális interfészen keresztül a **/var/log/pflog** fájlba. A PF a fájl feldolgozása során nem áll meg itt, az első passzoló szabálynál,

hanem végigolvassa a fájlt, s az alapján dönti el, hogy mi legyen a csomag sorsa. Kivétel itt is van, mert ha a quick kulcsszót találja, akkor nem olvassa tovább a konfigurációs fájlt, érvényre juttatja a szabályban foglaltakat. Tehát egyelőre tiltunk.

```
# loopback packets left unmolested
pass in quick on $Loop all
pass out quick on $Loop all
```

A saját gép visszacsatolási interfészén viszont mindent forgalom mehet...

```
# Immediate blocks
# fuzz any 'nmap' attempt
block in log quick on $Ext inet proto tcp from any to any flags FUP/FUP
block in log quick on $Ext inet proto tcp from any to any flags SF/SFRA
block in log quick on $Ext inet proto tcp from any to any flags /SFRA
```

Ezek a blokkolások megakadályozzák bizonyos hálózati kutakodóprogramok (nmap) eredményességét. FIXME Ezek mit is csinálnak _pontosan_? FIXME Ezen (és a többi tiltó) a szabályok minden blokkolt csomagot (forrás és cél IP címekkel, portokkal) naplóznak (erre a **log** szócska kényszeríti a **pf**-t), ami egy opcionális lehetőség. Sok értelme talán nincs, én szeretem látni a rosszalkodásokat egy fájlban.

```
# don't allow anyone to spoof non-routeable addresses
block in log quick on $Ext from $NoRoute to any
block out log quick on $Ext from any to $NoRoute
```

A nem route-olható címekkel való visszaélést is tiltani kell! FIXME

```
# silently drop broadcasts (cable modem noise)
block in quick on $Ext from any to 255.255.255.255
```

A teljes üzenetszórást is letiltjuk, s ez még naplózásra sem kerül.

```
# PASS rules
```

Most, hogy szinte mindent letiltottunk, jöhetnek azok, amiket mégis átengedünk.

```
# ALL  -- we don't normally do that. For debugging only.
#pass out quick on $Ext all keep state
```

Ez megjegyzésben marad, csak azért van itt, hogy ha valamilyen hibakeresés miatt mindent meg akarunk engedni, akkor ez könnyen megvalósítható legyen.

```
# pass in data mode connections for ftp-proxy running on this host.
pass in quick on $Ext inet proto tcp from any to any port > 49151 flags S/SA
keep state
```

Kiengedjük a korábban beállított ftp-proxy program általi forgalmat.

```
# ICMP
pass out     quick on $Ext inet proto icmp all icmp-type 8 code 0 keep state
pass in  log quick on $Ext inet proto icmp all icmp-type 8 code 0 keep state
```

Engedjük meg az ICMP forgalmat. Ez a (tűzfal)gépünk pingelését és traceroute-olását teszi lehetővé.

```
# Services we provide to the outside world
#pass in quick on $Ext inet proto udp from any to any port $InServicesUDP keep
state
pass in quick on $Ext inet proto tcp from any to any port $InServicesTCP flags
S/SA keep state
```

A korábban megadott portokon megengedjük a külvilágnak, hogy a tűzfalgépünkre csatlakozzon, annak szolgáltatásait használja. Ismétlem, vegyünk minél kevesebb portot fel az $InServicesTCP makróba a biztonság miatt.

```
# Standard services we want to access in the world
pass out quick on $Ext inet proto udp from any to any port $OutServicesUDP
keep state
pass out quick on $Ext inet proto tcp from any to any port $OutServicesTCP
flags S/SA modulate state
```

Azon szolgáltatások portjait engedjük kifelé, amelyeket használni szeretnénk bentről. A szabályoknál fontos a „keep state” opció, mert ezzel követi a kimenő kapcsolatot, s beengedi az erre adott választ. FIXME

```
# Special services
pass out quick on $Ext inet proto tcp from any to any port $XMMS flags S/SA
modulate state
pass out quick on $Ext inet proto tcp from any to any port $RealAudio flags
S/SA modulate state
```

Külön engedjük ki az xmms (webrádiók) által használt portokat.

```
# CVSup to cvsup.hu.freebsd.org
pass out quick on $Ext inet proto tcp from any to $CVSupServers port
$CVSupPorts flags S/SA modulate state
```

Otthoni gépemen néha FreeBSD-t indítok. A tűzfal és a FreeBSD kliensek frissítéséhez kell ez a szabály. Ez a CVSup portját negedi ki, de csak a megadott szerverek felé.

```
# Time server we use: time.kfki.hu (148.6.0.1)
pass out quick on $Ext inet proto tcp from any to 148.6.0.1 port time flags
S/SA modulate state
```

Időbeállításhoz használt időkiszolgáló a **time.kfki.hu**. Ha több szervert is el akarunk érni, akkor betehetnénk egy listába, vagy makróba, netán táblába, de most konkrétan adtam meg.

```
# Hungarian IRC servers:
pass out quick on $Ext inet proto tcp from any to $IRCservers port $IRCports
flags S/SA modulate state
# FreeNode IRC servers:
pass out quick on $Ext inet proto tcp from any to $FreeNodeIRCServers port
$IRCports flags S/SA modulate state
```

IRC eléréshez használt szabályok a fájl elején megadott címekre és portokon.

```
# FreeDB access:
pass out quick on $Ext inet proto tcp from any to $FreeDBhosts port
$FreeDBports flags S/SA modulate state
```

Internetes CD adatbázis elérése ily módon engedélyezhető.

```
# Jabber communication:
pass out quick on $Ext inet proto tcp from any to any port $JabberPorts flags
S/SA modulate state
```

Jabber – egy újabb azonnali üzenetküldési lehetőség.

```
# DynDNS.org update
pass out quick on $Ext inet proto tcp from any to $DynDNSServer port
```

```
$DynDNSPorts flags S/SA modulate state
```

A már leírt Dinamikus DNS frissítését engedi.

```
# BitTorrent connections
# pass out quick on $Ext inet proto tcp from any to any port { 6880><6889,
6969 } flags S/SAFR keep state
# pass in quick on $Ext inet proto tcp from any to any port 6880><6889 flags
S/SAFR keep state
```

E sorok a BitTorrent fájlcserélő rendszer szabályai lennének, de nem használjuk itt, mert külön, anchor-ral (horgonnyal) fogjuk szükség esetén betölteni.

```
# Other pass rules with anchor...
anchor passin
```

Itt a horgonyunk a további szabályokhoz. A nat.conf-nál már tárgyalt módon itt is lehet utólag szabályokat beilleszteni.

Ezzel a szabályokat áttekintettük. Értelmezésükhöz és finomításukhoz – még egyszer kérem – olvassuk el mindenképpen az OpenBSD oldalán a „PF User's Guide”-ot!!!

Üzemeltetés

Az üzemeltetés nem sok figyelmet igényel miután „beállt” a rendszer. Rendszerint a tűzfalszabályok frissítése (IP-k követése), naplók nézegetése és értelmezése, hibajavítások alkalmazása, stb. tartozik ide.

1 Tűzfalszabályok frissítése

Ez általában akkor szükséges, ha az egyik általunk használt kiszolgáló IP címe megváltozik. Nálam ez a freenode IRC szervereit szokta jelenteni, vagy egyéb „bedrótozott” kiszolgáló címe. Ilyenkor átírom a tűzfalon a **/etc/pf.conf** fájlt, s újra betöltöm:

```
pfctl -Rf /etc/pf.conf
```

Ha a NAT szabályai is módosulnak, akkor azt is alkalmazni kell:

```
pfctl -Nf /etc/nat.conf
```

2 Belső gép elérése ssh-val

Ha egy belső gépet szeretnénk kintről elérni, egy átirányítást kell megadnunk a tűzfalunk számára. Ezt úgy (is) meg lehet oldani, hogy kiválasztunk egy általában nem használt portot, s az arra irányuló kéréseket átirányítjuk a belső gép 22-es portjára. Ez a **nat.conf**-ban így jelenik meg:

```
rdr on tun0 proto tcp from any to any port 5859 -> 192.168.0.10 port 22
```

E szabály betöltése után ha valaki a tűzfalunk 5859-es portjára ssh-zik, akkor a 192.168.0.10-es gépre próbál belépni...

Szabály betöltése:

```
pfctl -Nr /etc/nat.conf
```

3 Naplóelemzés

FIXME

A legfontosabb naplófájl a **/var/log/messages**. Ide naplózza a gép a legtöbb problémát és információt.

4 Hibajavítások alkalmazása

Nagyon fontos erre figyelni! Az OpenBSD errata oldalát (http://www.openbsd.org/errata.html) érdemes gyakran látogatni, s ha új hibajelentés (errata) jelenik meg, azt érdemes mielőbb alkalmazni a rendszerünkre.

Hogyan működik ez?

- Megszerezzük a legfrissebb forrást,
- Elvégezzük a frissítést .

FIXME

Forrás megszerzése

A forrás frissentartása alapvetően kétféleképpen mehet: cvsup-pal illetve patch-ekkel.

Az egyszerűbb mód a patch-elés. Ehhez le kell tölteni a kiadás forrását. Ha a telepítő CD létrehozása pontnál ezt már megtettük, most nem kell. Csomagoljuk ki a forráscsomagokat a /usr/src könyvtárba:

```
cd /usr/src
tar xvfz /ahol/a/forrasfajlok/vannak/src.tar.gz
tar xvfz /ahol/a/forrasfajlok/vannak/sys.tar.gz
```

Ha az Xfree-t is telepíteni akarjuk:

```
tar xvfz /ahol/a/forrasfajlok/vannak/XF4.tar.gz
```

Szóval, ha megvan az alap forrás, hozzá lehet adni a javítófoltokat. Ezeket megtaláljuk itt: ftp://ftp.openbsd.org/pub/OpenBSD/patches/, konkrétan a 3.5-ös verzióhoz tartozó **3.5.tar.gz** fájlt. A letöltött és kicsomagolt fájlokat a **patch** parancssal lehet alkalmazni a „szűz” forrásfára...

Kicsomagolás:

```
cd /usr/src
tar xvfz /ahol/a/forrasfajlok/vannak/3.5.tar.gz
```

A patch-ek egy saját könyvtárfát alkotnak a verziószámmal legfelül, s alatta a közös (common) illetve az egyes architektúrákra vonatkozó alkönyvtárakkal.

Egyenként célszerű alkalmazni őket valahogy így:

```
patch -p0 < nagyonfontos1.patch
patch -p0 < nagyonfontos2.patch
```

Egy barbár módszer az összes patch alkalmazására:

```
find /usr/src/3.5 -name \*\.patch -exec patch -p0 < {} \;
```

A forrás megszerzésének másik (javasolt) módja a cvsup frissítés. Ehhez egy supfile-t hozunk létre, amit paraméterként megadva a **cvsup** programnak frissíti a forrásfá(kat). Persze ehhez a **cvsup** programnak telepítve kell lennie. A telepítés menete:

```
cd /usr/ports/net/cvsup
setenv FLAVORS no_x11
make install
```

Ha a telepítés nem megy forrásból, akkor töltsük le az előre lefordított bináris csomagot, s telepítsük azt (ezt ajánlja az OpenBSD CVSup FAQ is).

A program telepítése után szülessen meg a konfigurációs fájl. Én kettébontottam a letöltést a ports fára és a rendszer fára, ezért két supfile-t hoztam létre. A port fa supfile-ja:

/etc/supfile.cvs

```
# $Id: supfile.cvs,v 1.2 1997/01/17 05:11:44 jdp Exp $

*default host=cvsup.hu.openbsd.org compress
*default release=cvs tag=OPENBSD_3_5
*default base=/usr
*default prefix=/usr
*default delete use-rel-suffix compress
*default tag=OPENBSD_3_5
#src-all
#src-eBones
#src-secure
#ports-all
OpenBSD-ports
```

Ebben megadom a szerver nevét, majd a rendszer verzióját. Ezután a könyvtárak specifikálása történik és a legvégén megadom, hogy csak a ports-fát szedje le.

A rendszer forrását e fájl segíti az én gépemre (a fájl neve bármi lehet):

```
# $Id: supfile.cvs,v 1.2 1997/01/17 05:11:44 jdp Exp $

*default host=cvsup.hu.openbsd.org compress
*default release=cvs tag=OPENBSD_3_5
*default base=/usr
*default prefix=/usr
*default delete use-rel-suffix compress
*default tag=OPENBSD_3_5
#src-all
#src-eBones
#src-secure
#ports-all
OpenBSD-src
#OpenBSD-ports
#OpenBSD-www
#OpenBSD-all
```

Most, hogy megvannak a segédfájlok, mehet a frissítés:

```
cvsup -g -L 7 /etc/supfile.cvs
cvsup -g -L 7 /etc/supfile.allsrc
```

Vigyázat, a művelet sokáig tart, főként az első alkalommal, ha nincs még semmi a forrásfában! A ports fa a /usr/ports alatt, míg a rendszer forrása a /usr/src alatt frissül.

Frissítés alkalmazása

A frissítés úgy történet, hogy az egyes patch-ekből kiolvassuk (az elején a megjegyzésekből), melyik az a rész, amit újra kell fordítani. Például az OpenBSD 3.5 013-as

patch-e a következőket „írja elő”:

```
Apply by doing:
        cd /usr/src
        patch -p0 < 013_httpd.patch

And then rebuild and install httpd and its modules:
        cd usr.sbin/httpd
        make -f Makefile.bsd-wrapper obj
        make -f Makefile.bsd-wrapper cleandir
        make -f Makefile.bsd-wrapper depend
        make -f Makefile.bsd-wrapper
        make -f Makefile.bsd-wrapper install

If httpd had been started, you might want to run
        apachectl stop
before running "make install", and
        apachectl start
afterwards.
```

Lépésről lépésre megadja, hogyan frissítsük a kompromitált programot (jelen példában az apache httpd webszervert). Sokszor új kernelt kell fordítani (Lásd a *Kernelfordítás* pontot).

5 Kernelfordítás

A kernelfordításról mindenképpen szó kell, hogy essen! Szükség lehet rá egyes hibajavítások (errata) megjelenésekor, egyéb esetekben FIXME (mikor is ?:). Az OpenBSD csapat nem javasolja a saját kernel fordítását, sőt, a GENERIC (a fejlesztők által ajánlott konfigurációjú) kernel esetén támogatás sem jár (egyébként igen FIXME?).

Tehát mi most egy GENERIC kernelt fogunk fordítani (mondjuk egy errata hibajelentésének hatására). Ehhez meg kell lennie a legújabb, vagy a kívánt verziójú, vagy a patchelt kernelforrásnak. Ennek megszerzését/frissítését leírtam a *Forrás megszerzése* szakaszban.

Belépünk a kernel konfigurációs fájljának könyvtárába:

```
cd /usr/src/sys/arch/i386/conf
```

Az itt lévő GENERIC fájl tartalmazza az alapértelmezett kernelkonfigurációt. A javasolt mód a GENERIC fordítása, de ha — jó meggondolásból — mégis saját kernelt akarunk, akkor más néven elmentve átszerkesztjük (például kivesszük az SCSI meghajtókat, ha erre nincs szükségünk).

Ezután jöhet a kernel konfigurálása:

```
config GENERIC
```

Belépünk a kernel forráskönyvtárába:

```
cd ../compile/GENERIC
```

Jöhet a fordítás:

```
make depend
make
```

Ez elég sokáig tart, keressünk valami jó elfoglaltságot... Ha kész, és nem volt hibaüzenet, az aktuális könyvtárban létrejött az új kernel binárisunk, **bsd** néven, amit átmásolhatunk a gyökérbe (az eredetit elmentve):

```
cp /bsd /bsd.old
```

```
cp ./bsd /bsd
```

Újraindítás után az új kernelünk nagy valószínűséggel bebootol, s minden hardverünket sikeresen felismeri.

Ha mégsem indulna el, akkor bootoljunk a régi kernellel. A boot promptkor a következőt kell begépelni:

```
boot /bsd.old
```

6 Operációs rendszer teljes frissítése

Az időnkénti kiadások/verzióváltások mindig új dolgokat hoznak, jó és néha rossz új dolgokat. Az ember mindig bízik abban, hogy a jóból sokkal több van. Az eljárásnál kövessük az OpenBSD honlap „Upgrading Mini-FAQ” oldalát:

1 Az első lépés a legfrissebb forrás beszerzése a korábban már leírt módon cvsup-pal vagy letöltés+patch-elés kellő kombinációjával.

2 Ezután kitakarítjuk a régi object fájlokat a következő parancsokkal:

```
cd /usr/src
ind . -type l -name obj | xargs rm
make cleandir
rm -rf /usr/obj/*
make obj
```

3 Itt következik egy fontos lépés, mégpedig az új verzióhoz tartozó kézi változások végrehajtása. Például új felhasználó létrehozása. Ezt mind jól leírja a http://www.openbsd.org/faq/upgrade-minifaq.html dokumentum az egyes verziók közötti váltásra vonatkozó szakasza. Például a http://www.openbsd.org/faq/upgrade-minifaq.html#3.4 oldal a 3.4-ről 3.5-re váltás közti lépéseket írja le. Ha ezt kihagyjuk, sok kellemetlenséget veszünk a nyakunkba FIXME.

4 A következő parancs létrehozza az esetleges új könyvtárakat:

```
   cd /usr/src/etc && env DESTDIR=/ make distrib-dirs
```

5 Új kernel fordítása:

```
cd /usr/src/sys/arch/`machine`/conf
config GENERIC
cd ../compile/GENERIC
make clean && make depend && make
cp /bsd /bsd.old && cp bsd /bsd
```

6 Rendszer fordítása:

```
cd /usr/src
make build
```

Ez a lépés igen-igen hosszas. Türelem.

Ha hiba történik, eléggé magunkra maradtunk, de a hibaüzenetet vizsgálgatva esetleg kitalálható, mi volt a gond. Próbálkozhatunk a „make build” parancs újra futtatásával. Általában azért lefut gond nélkül, kiváltképp akkor, ha a verzióváltás által előírt változtatásokat rendben végigvisszük, illetve a forrásunk tiszta, azaz a RELEASE-fát patch-eltük meg vagy üres könyvtárba töltöttük le a forrást cvsup-pal FIXME.

7 Bittorrent beállítása

A Bittorrent átengedéséhez elég sok portot szabadon kell hagynunk. Ezért csak ún. horgonyokkal engedjük meg akkor, amikor használni szeretnénk. Állandóan nem kell engedni — szerintem.

A horgonyt már letettem a /etc/pf.conf és a /etc/nat.conf végére. Ez egy név, amire hivatkozunk majd, s a horgony helyébe illeszti az utólagos szabályt a pf. Mivel a lustaság fél egészség, ezért ezt néhány fájl és szkript segítségével oldom meg. A beillesztendő szabályokat a /etc/pf.<szabálytípus> és /etc/nat.<szabálytípus> fájlokba teszem így:

/etc/pf.torrent

```
Ext=tun0
# BitTorrent connections
pass out quick on $Ext inet proto tcp from any to any port { 6880><6889,
6969 } flags S/SAFR keep state
pass in quick on $Ext inet proto tcp from any to any port 6880><6889 flags
S/SAFR keep state
```

/etc/nat.torrent

```
Ext="tun0"
rdr on $Ext proto tcp from any to any port 6881:6889 -> 192.168.0.10 port
6881:*
```

Betöltéshez a pfctl programot kell használni, de ez szkriptesítve lett. A szkriptnek két paramétere van, az első a szabály típusa, a második a start vagy stop szavak valamelyike attól függően, hogy beilleszteni vagy kivenni szeretnénk.

/usr/local/sbin/pfa

```
#! /bin/sh
#
#
if [ $# -ne 2 ] ; then
    echo " Usage: `basename $0` <ruletype> start|stop"
    exit 7
fi
if [ ! -f /etc/pf.$1 -a ! -f /etc/nat.$1 ] ; then
    echo " The files for $2 ruletype don't exist!"
    exit 8
fi

case $2 in

 start )
    echo "Enabling $1 ..."
    pfctl -vv -a passin:$1 -f /etc/pf.$1
    pfctl -vv -a redirect:$1 -f /etc/nat.$1
    echo "Done."
 ;;

 stop )
    echo "Disabling $1 ..."
    pfctl -a passin:$1 -F rules
    pfctl -a redirect:$1 -F nat
 ;;

 * )
    echo " usage: `basename $0` <ruletype> start|stop"
    echo
 ;;
esac
```

Például:

```
tuzfalam# pfa torrent start
Enabling torrent ...
Loaded 293 passive OS fingerprints
Ext = "tun0"
@0 pass out quick on tun0 inet proto tcp from any to any port 6880 >< 6889
flags S/FSRA keep state
@1 pass out quick on tun0 inet proto tcp from any to any port = 6969 flags
S/FSRA keep state
@2 pass in quick on tun0 inet proto tcp from any to any port 6880 >< 6889
flags S/FSRA keep state
Loaded 293 passive OS fingerprints
Ext = "tun0"
@0 rdr on tun0 inet proto tcp from any to any port 6881:6889 -> 192.168.0.10
port 6881:6889
Done.
```

```
tuzfalam# pfa torrent stop
Disabling torrent ...
rules cleared
nat cleared
```

Ha mégis parancssorból óhajtjuk ezeket berakni illetve kivenni, akkor ez a következő parancsokkal vihető végbe. A sorrend fontos, először az átengedési, majd a címfordítási szabály kell következzen.

Beillesztés:

```
pfctl -v -a passin:torrent -f /etc/pf.torrent
pfctl -v -a redirect:torrent -f /etc/nat.torrent
```

Törlés:

```
pfctl -v -a passin:torrent -F rules
pfctl -v -a redirect:torrent -F nat
```

8 Belső Windows XP adminisztrálása RDP-n keresztül

FIXME

„- /etc/nat.conf-hoz az RDP portja 3389 (ha valaki belső hálón lévő WinXP-t akar távolról adminisztrálni)” -- M. E. után

9 Windows-os gép elérése RDP protokollon keresztül

Az újabb Windows rendszerek kínálnak egy VNC-hez hasonló jellegű szolgáltatást, az asztal megosztást / terminál szerver szolgáltatást vagy képernyő átvételt (FIXME hogy is hívják ezt odaát?). Ezt az ún. RDP protokollon keresztül valósítja meg. Ennek engedélyezését a Bittorrenthez hasonlóan ad-hoc jelleggel is meg lehet oldani:

/etc/pf.rdp

```
Ext=tun0
# Terminal server (RDP)
pass out quick on $Ext inet proto tcp from any to any port 3389 flags S/SAFR
keep state
```

Ezután hozzunk létre egy üres fájl /etc/nat.rdp néven:

```
touch /etc/nat-rdp
```

Ezután jöhet az RDP kiengedése a tűzfalon:

```
tuzfalam# pfa rdp start
Enabling rdp ...
Loaded 293 passive OS fingerprints
Ext = "tun0"
@0 pass out quick on tun0 inet proto tcp from any to any port = 3389 flags
S/FSRA keep state
Loaded 293 passive OS fingerprints
Done.
```

Ha már nem kell, kikapcsolhatjuk:

```
tuzfalam# pfa rdp stop
Disabling rdp ...
rules cleared
nat cleared
```

Amennyiben állandóan szükség lehet erre a szabályra a /etc/pf.rdp „pass” kezdetű sorát tegyük be a /etc/pf.conf vége felé valahova.

10 Ha nem jól működik a BackSpace gomb

Ha belépünk a tűzfalunkra, bizonyos terminháltípusoknál gond lehet a Backspace billentyű leütésekor. Ha visszatörlés helyett „^?” jeleket küld adjuk ki a következő parancsot:

```
stty erase ^?
```

A „^?” jeleket a Backspace billentyű leütésével bigyük be!

Kapcsolódó dokumentumok

BSD partíciók és slice-ok: http://www.bsd.hu/dokumentacio/bevezetes/slice/view

Egyéni, vagy sok hasonló rendszer telepítése:
http://www.openbsd.org/faq/faq4.html#Multiple

Én főként ez alapján csináltam: http://www.realo.ca/BSDinstall.html

OpenBSD - HUP Wiki bejegyzés:
http://www.hup.hu/wiki/wiki.phtml?title=OpenBSD&PHPSESSID=f975a2ffb3ae0db7119f9ad10bb07dd3

UNIX történelem:
http://www.hup.hu/modules.php?name=Sections&op=viewarticle&artid=28
http://www.hup.hu/modules.php?name=Sections&op=viewarticle&artid=29
http://www.hup.hu/modules.php?name=Sections&op=viewarticle&artid=31

BSD történelem:
http://www.hup.hu/modules.php?name=Sections&op=viewarticle&artid=40
http://www.hup.hu/modules.php?name=Sections&op=viewarticle&artid=41
http://www.hup.hu/modules.php?name=Sections&op=viewarticle&artid=42
http://www.hup.hu/modules.php?name=Sections&op=viewarticle&artid=43

NAT - HUP Wiki bejegyzés: http://www.hup.hu/wiki/wiki.phtml?title=NAT

Papamike szuper OpenBSD leírásai (ADSL, tűzfal, DNS):
http://www.aei.ca/~pmatulis/main_menu.html

BitTorrent beállítása (anchor-ral)
http://unix.derkeiler.com/Newsgroups/comp.unix.bsd.openbsd.misc/2003-12/0046.html

http://www.fmi.uni-passau.de/~grafj/openbsd/3.4/#11.3_BitTorrent

ddclient DynDNS kliens http://s90389134.onlinehome.us/ddclient/ddclient.tar.gz

PF dokumentáció http://www.openbsd.org/faq/pf/index.html

ALTQ – forgalomszabályozás ADSL-nél: http://www.benzedrine.cx/ackpri.html

OpenBSD errata: http://www.openbsd.org/errata.html

OpenBSD Upgrading Mini-FAQ: http://www.openbsd.org/faq/upgrade-minifaq.html

OpenBSD CVSup FAQ: http://www.openbsd.org/cvsup.html

Fairly-Secure Anti-SPAM Gateway Using OpenBSD, Postfix, Amavisd-new, SpamAssassin, Razor and DCC http://www.flakshack.com/anti-spam/