Csatlakozás a Hálóra Linuxszal (2. rész)

Az előző részben összeállítottunk egy működő útválasztót a Coyote Linux-változat segítségével. Ebben a részben az útválasztó további beállításairól, távoli karbantartásáról és bővítéséről lesz szó.

Először is nem árt ismerni az útválasztó működési elvét. Mivel a szolgáltatótól csak egy valós (Interneten használható) IP-címet kapunk, belső hálózatunk gépei „saját nevükben" nem tudnak az Internetre csomagot küldeni és onnan fogadni. Az útválasztó ezt úgy oldja meg, hogy az úgynevezett NAT (Network Address Translation: hálózati címváltás) egyik módszerét használja: az IP-álcázást (masquerading). Az álcázás úgy valósul meg, hogy a belső hálózat gépei által küldött csomagokat az útválasztó a saját IP-címével egy eltérő kapuról küldi ki az Internetre, és ugyanígy fogadja a választ, majd a belső IP-címet visszahelyezve továbbítja azt a belső hálózat gépére. Egy táblázatot tart fenn annak érdekében, hogy ezeket az IP-cím-átváltásokat el tudja végezni. A belső gépek úgy látják, mintha közvetlenül az Interneten lévő géppel lennének kapcsolatban, az Internet felől pedig úgy tűnik, mintha csak az útválasztóval tartanának kapcsolatot. Itt hívnám fel a figyelmet rá, hogy elvben megközelítőleg megállapíthatjuk, hogy hány gép működik az útválasztó mögött „rejtve" (lásd ***Steven M. Bellovin*** cikkét). Ezért, ha a szolgáltatónk kifejezetten tiltja a több géppel történő használatot, akkor, ha meg is építjük az útválasztót, ne nagyon használjuk több géppel egy időben (csak az Internetre irányuló forgalom számít). Én azon az állásponton vagyok, hogy ha a felhasználási szerződés kifejezetten nem tiltja, akkor szabad a pálya. (Mivel nincs benne, hogy csak egy adott operációs rendszerrel lehet igénybe venni, ugyanekkora terhelést el lehetne érni egy linuxos, illetve unixos gépen, és arra a helyi hálózaton bejelentkező felhasználók esetén is.)

Hibalehetőségek a telepítés során

Biztosan többféle hiba merülhet fel a telepítés után. Egy elég részletes GYK (FAQ) található meg hosszas keresés után a ➲ http://www.dalantech.com oldalain. Itt egyébként rendkívül széles körű, igaz, angol nyelvű technikai adathalmaz található. Egy hibát említenék, ami elsőre eléggé nehezen volt felfedezhető. A modemes behívásnál egy úgynevezett chatprogram veszi fel a kapcsolatot először a szolgáltató gépével. Miután ez befejezte, akkor indul el a `ppp` program; sok esetben ez a program kezeli a bejelentkezést is. Ennek kérését a kiszolgáló általában egy jelszó-készenlétijel (`login:` prompt) küldésével jelzi. A mi esetünkben azonban ehelyett `username` készenléti jelet küldött, így a chatprogram hiába várt a jelszóablakra – ezt tehát módosítani kellett.

Több módszer is akad a Coyote rendszer módosítására. Az egyik, hogy billentyűzetet és monitort csatlakoztatunk a géphez, és bejelentkezünk. Lényegében ugyanez a módszer akkor is, ha `ssh` segítségével már megy a távoli bejelentkezés. Csak egyetlen felhasználó van: a rendszergazda. Az első bejelentkezéskor adhatunk meg jelszót. Bejelentkezés után az `lrcfg` program indul el. Ebből a `q` ponttal léphetünk ki. Ha nem tudjuk, melyik könyvtárban vagyunk (a rendszer nem írja ki a készenléti jelnél), a `pwd` paranccsal megtudhatjuk. A fent említett hiba kiküszöböléséhez lépjünk a */etc/ppp* könyvtárba, és itt szerkesszük a ***szolgaltatonev.chat*** fájlt. (A szolgáltatónevet a lemez készítése közben mi adtuk meg. Ezért kell rövidnek és ékezet nélkülinek lenni.) A Coyote rendszernek része az `ae` szövegszerkesztő. Ezt az `ae fÆjlnØv` paranccsal elindítva elvégezhetjük a szükséges módosításokat. Az F1 lenyomásával kaphatunk segítséget a billentyűparancsokhoz. Ebben az esetben az `ogin:` részt kell kicserélni a `sername:` bejegyzésre. (Azért nincsenek ott az első betűk, mert nem lehet tudni, hogy kis- vagy nagybetűvel kezdve küldi-e a kiszolgáló.)

Miután módosítottunk a rendszeren, tudnunk kell, hogy a módosítás csak a memórialemezen következett be, azaz újraindítás esetén elvész, ezért mielőtt újraindítanánk, hajlékonylemezre kell mentenünk. Ha átkapcsoltuk az írást tiltó fület, akkor újra engedélyezzük. (A végső beállítás után nem árt csak olvashatóvá tenni, így egy esetleges betörés esetén se tudnak olyan kárt tenni az útválasztóban, amit egy újraindítás ki ne küszöbölne.) Indítsuk el a `lrcfg` programot. Ebben válasszuk a ***b*** pontot. A mentés hamarosan elkészül. Ezek után újabb indításkor már a módosított állományt használja a gép.

Pár szó a Coyote felépítéséről

A különböző feladatokhoz szükséges állományokat tgz- (***tar.gz***) fájlokban, tömörítve tárolja a hajlékonylemezen. Ezzel (is) magyarázható az, hogy ennyi minden elfér egyetlen lemezen. Indításkor önműködően létrehoz egy memórialemezt, és ebbe csomagolja ki a szükséges állományokat – így alakul ki a működő fájlrendszer.

Ebből adódik a másik módosítási lehetőség. Az útválasztó lemezét saját linuxos munkaállomásunkon is módosíthatjuk. Fűzzük be a hajlékonylemezt a rendszerbe, majd másoljuk a megfelelő ***.tgz***-fájlt (a fenti feladatnál ez az ***etc.tgz*** volt) egy üres könyvtárba, ott csomagoljuk ki a `tar xvzpf fÆjlnØv` paranccsal. Ezután a létrejövő könyvtárakban és fájlokban elvégezhetjük a szükséges módosításokat; utána a `tar cvzpf fÆjlnØv` paranccsal csomagoljuk vissza az állományokat, majd másoljuk vissza őket a lemezre.

Egy jó tanács: ha működő rendszerünk van, akkor a módosítgatások előtt célszerű lenyomatot készíteni róla, hogy ha valami rosszul sülne el, pótlemezt készíthessünk. A szükséges lenyomat a

```
dd if=/dev/fd0u1680 of=lenyomatnev.fÆjl
```

paranccsal készíthető el. Ebből hajlékonylemezt az `if` és `of` kapcsolók cseréjével készíthetünk.

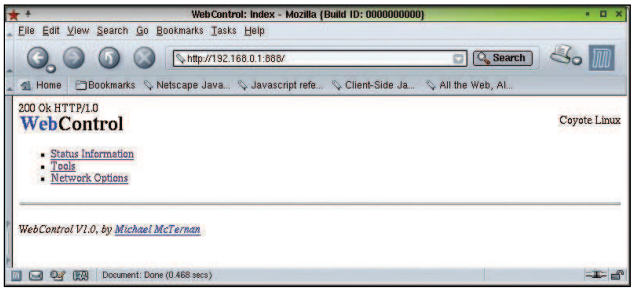
Távoli karbantartás

Előfordulhat, hogy a munkaállomásokon is tudnunk kell az útválasztó IP-címét. Az egyik kézenfekvő példa erre az, ha azt szeretnénk, hogy valamelyik (feltehetően Linuxot futtató) szeretnénk,

hogy valamelyik (feltehetően Linuxot futtató) munkaállomás webkiszolgálóként működjön, és így lehetővé váljon, hogy bár változó IP-címmel, de teljes dinamikus valójában megmutathassuk webes munkánkat másoknak. Ehhez azonban meg kell tudnunk mondani, hogy az útválasztó éppen milyen IP-címen érhető el, és emellett a kívánt HTTP-kaput át kell irányítanunk az adott munkaállomásra (erről az utóbbiról később lesz szó).
Ha telepítettük a `webctl` csomagot, akkor viszonylag könnyen megtudhatjuk a hálózati adatokat. Írjuk be a böngészőbe (Mozilla ajánlott, a régebbi Netscape-pel gondok akadtak a megjelenítéssel során):

192.168.0.1:888

Azért nem a szabványos 80-as kapun működik, hogy azt át tudjuk irányítani. Ekkor a *képen* látható egyszerű felület jelenik meg.



Innen a ***Status Information*** hivatkozásra (link) kattintva részletes tájékoztató oldalt kapunk, gyakorlatilag az `ifconfig`, `route`, és az `ipchains -L` parancsok webesített változatát. A ***Tools*** hivatkozást követve a naplófájlok tartalmát nézhetjük meg, valamint egy egyszerű névfeloldást használhatunk (DNS lookup). A ***Network options***-t választva engedélyezhetjük a `telnet`-elérést, tilthatunk, illetve megnézhetjük a tűzfal szabályait. Elvileg hozzá is lehet adni, de ez a lehetőség egyelőre nem működik.
A másik lehetőség a távoli karbantartásra az SSH alkalmazása. Ha a Coyote-lemez elkészítésekor kiválasztottuk ezt a csomagot, akkor fent van az útválasztón. (Az útválasztóra bejelentkezve és a `ps` parancsot kiadva egy */usr/sbin/sshd* folyamatot kell látnunk). Ahhoz azonban, hogy egy munkaállomásról be tudjunk lépni, egy kulcspárt kell készítenünk, és a nyilvános részét fel kell telepítenünk a Coyote-ot futtató gépre. Figyelem, ha már van `ssh` kulcspárunk (mert más célra készítettünk már egyet), és használjuk is, akkor az alábbi parancsokat ne futtassuk alapértelmezett módon, mert felülírja a régi kulcsot. Ha régebbi, 1-es változatú SSH-nk van, akkor elég az alábbi parancsot a saját könyvtárunkban kiadni:

```
ssh-keygen
```

Ha a Linuxunk frissebb, és már van rajta SSH2, akkor meg kell adnunk azt is, hogy milyen fajta kulcsokat készítsen. Mivel a Coyote SSH1-t használ, így ennek megfelelő kulcsot készítünk az alábbi paranccsal:

```
ssh-keygen -t rsa1
```

Megkérdezi, hogy hová mentse a kulcsot – ha nincs másik kulcsunk, hagyjuk az alapértelmezett értéken. Ezután (kétszer) be kell írnunk egy jelszót, ami a kulcsunkat védi, hogyha valaki le tudja is másolni a gépünkről, akkor se tudja enélkül használni. Célszerű tehát beírni valamit (üres kulcsot is megenged). A rendszerjelszóval ellentétben szóközt is tartalmazhat, és hosszabb szöveg is begépelhető.
Ha elkészült, akkor nincs más dolgunk, mint átmásolni az útválasztóra. Mivel az SSH még nem működik, ilyen módon vagy a fentebb említett rendszerlemez teljes Linux-rendszerünkhöz való csatlakoztatásának módszerét alkalmazhatjuk, vagy pedig egyszerűen másoljuk ki a kulcs nyilvános felét tartalmazó *~/.ssh/identity.pub* fájlt egy hajlékonylemezre. Ezután a lemezt a Coyote rendszeren az alábbi parancsokkal csatlakoztassuk, és másoljuk be a kulcsot a megfelelő helyre:

```
cd /
mkdir /floppy
(tegy k be a lemezt a kulccsal)
mount /dev/fd0 /floppy
mkdir ~/.ssh
cp /floppy/identity.pub ~/.ssh/authorized_keys
umount /floppy
```

Ezek után, ha minden jól ment, készen állunk a próbára. A munkaállomáson, amelyről a kulcsot vittük, a következő paranccsal próbálkozzunk:

```
ssh 192.168.0.1
```

Megkérdi, hogy felvegye-e a 192.168.0.1 kiszolgálót az ismert gépek (known hosts) listájára. A teljes `yes` szó beírásával kell válaszolnunk. Ezután a jelszót (passphrase) kéri, írjuk is be.
Ha minden jól ment, azon nyomban látjuk az `lrcfg` program menüjét, és benn is vagyunk az útválasztón. Még az van hátra, hogy az új beállítást mentsük. Ehhez előbb módosítanunk kell a */etc/coyote/packages* fájlt. Írjuk bele, hogy a `root` csomagot is mentse (a `root` szót tartalmazó sor). Ezt követően az `lrcfg` programot elindítva és a ***b*** menüpontot választva menti a rendszert, és a következő indításkor már ezt használja. Ha elkészült, a fenti fájlt visszaírja az eredeti állapotra (a `root` sort kivéve), és újra menti a rendszert – a `root` csomagot a további mentésekből kizárhatjuk, és így jelentősen felgyorsíthatjuk őket.
Itt mindent ugyanúgy meg tudunk tenni, mintha közvetlenül az útválasztó billentyűzetét és monitorát használva jelentkeztünk volna be. Ha működik, nincs is szükség rájuk, le is vehetjük őket. (Ha esetleg nagyot hibázunk és elbabrálunk valamit, akkor lesz rájuk újra szükségünk.) Próbaként lefuttathatjuk az `ifconfig`, a `route` és a `dmesg` parancsokat – ezekből sokat megtudhatunk a rendszer állapotáról.

Az útválasztó finomhangolása

A rengeteg lehetőség közül kettőt ragadok ki. Az egyik a kapuváltás, hogy a külvilág számára egy belső munkaállomás valamilyen kiszolgálóként működhessen. A másik pedig több belső hálózat kezelése.
A kapuátirányítás elsőre rendkívül könnyűnek tűnik, mivel egy erre beépített rész található az IP-álcázás parancsfájljában. Ez azonban alaphelyzetben nem működik, sajnos bele kell javítanunk (valószínűleg változott a különböző változatok között).
A kívánt kapuváltásokat a */etc/coyote/portforwards* fájlban kell megadnunk, egy sorban egyet, a következő formában: kezdőkapu végsőkapu protokoll célgép.
Ha egy sort kettős kereszttel (#) kezdünk, azt megjegyzésként kihagyja. A különböző adatok között szóköznek és nem veszszőnek kell lennie. Ha a HTTP-kaput akarjuk egy belső gépre átirányítani, használhatjuk a következő sort:

```
80 80 tcp 192.168.0.5
```

A protokoll lehet UDP is, ha arra van szükség. Egy kapuprotokollpárt csak egy belső gépre lehet átirányítani.
Most van soron a parancsfájl javítása. Nyissuk meg a `/etc/rc.d/rc.masquerade` fájlt. A `set_port_forward` függvényt módosítsuk a következő formára:

```
set_port_forward() {
if ! [$# = 4] ; then
    echo " rvØnytelen kapuvÆltÆsi szabÆly a
    ➥/etc/coyote/portforwards fÆjlban!"
    return 1
    fi
    /sbin/ipmasqadm autofw -A -r $3 $1 $2
    ➥-h $4
    }
```

Ha ezek után le is futtatjuk a parancsfájlt (`/etc/rc.d/rc.masquerade`), akkor egy külső, Interneten lévő gépről ki is próbálhatjuk, hogy meg tudja-e nyitni a belső gépen lévő HTTP-kiszolgáló nyitólapját. Ha működik, akkor a szokott módon mentsük a rendszert (`lrcfg` ***b*** pont).
Most pedig bővítsük ki az útválasztót két belső hálózat kiszolgálására. Mindkettő számára biztosítja az IP-álcázást az internetéréshez, emellett – mint egy hagyományos útválasztó – össze is köti a két hálózatot, vagy ha a hálózati hordozó eltérő (például koax és sodrott érpár), akkor hídként. További szerepe lehet még a hálózat fizikai határának kitolása: tőle mindkét irányban a legnagyobb kábelhosszt lehet alkalmazni, így a két legtávolabbi gép az eredetileg lehetséges távolság kétszeresére lehet egymástól.
Ha a fenti példához hasonlóan a második belső hálózathoz ugyanolyan típusú hálózati csatolót alkalmazunk, mint az elsőhöz vagy mint az internetoldalihoz, és emellett a kártya PCI-sínes, akkor semmit sem kell tennünk az eszköz felismeréséhez, mert a modul az összes azonos típusú kártyát önműködően kezeli. Ha másfajta kártyát használunk, akkor a megfelelő modulokat a Coyote rendszer ***/lib/modules*** könyvtárába kell telepíteni, majd a ***/etc/modules*** fájlba kell beírni őket, ha szükséges, az értékekkel együtt. Ezután pedig az előzőekhez hasonlóan menteni kell a rendszert.
Újraindításkor figyeljük meg, hogy az összes kártya elindult-e. Ha igen, akkor már csak a program beállítása maradt hátra. Mivel ehhez nincs külön parancsfájl, kézzel kell megtennünk. Először is módosítsuk a ***/etc/coyote/coyote.conf*** fájl hálózati adatokkal kapcsolatos részét az alábbiak szerint:

```
LOCAL_IPADDR=192.168.0.1
LOCAL_NETMASK=255.255.255.0
LOCAL_BROADCAST=192.168.0.255
LOCAL_NETWORK=192.168.0.0
LOCAL_IPADDR2=192.168.1.1
LOCAL_NETMASK2=255.255.255.0
LOCAL_BROADCAST2=192.168.1.255
LOCAL_NETWORK2=192.168.1.0
```

Mint látható, egy második hálózati beállításcsomagot adtunk hozzá, és négy újabb meghatározást (ezek egyébként a héjváltozók). Ettől még nem működik a második hálózat; azért van rá szükség, hogy könnyebben be tudjuk írni a megfelelő értékeket a parancsfájlokba.
Először is állítsuk be a `/etc/rc.d/rc.inet` parancsfájlt. Ez indítja el a hálózati eszközök kezelőprogramját. Írjuk be a kisebb jellel (>) jelölt sorokat a jelöletlen sorok közé. (A > jel nélkül).

```
....
# Set the interfaces to use for Internet /
➥Internal Networks
if [ -z "$IF_LOCAL" ]; then
       IF_LOCAL=eth0
fi
> if [ -z "$IF_LOCAL2" ]; then
>      IF_LOCAL2=eth2
> fi
.....
# Configure the local network
ifconfig $IF_LOCAL $LOCAL_IPADDR netmask
➥$LOCAL_NETMASK broadcast $LOCAL_BROADCAST
if [ $? = 0 ]; then
       LOCAL_UP=YES
else
       LOCAL_UP=NO
fi

> ifconfig $IF_LOCAL2 $LOCAL_IPADDR2 netmask
➥$LOCAL_NETMASK2 broadcast $LOCAL_BROADCAST2

# Enable kernel level IP forwarding
echo 1 > /proc/sys/net/ipv4/ip_forward

...
```

Az `ifconfig` egy sorban helyezkedik el a `$LOCAL_BROADCAST-tal`. Ezzel a hálózati csatolók egy újraindítás után elindulnak, és mivel a magban engedélyezett az IP-csomagtovábbítás, a két hálózat között máris hídként működik. A második hálózat internetéréséhez még bele kell írnunk az `rc.masquerade` parancsfájlt. A következő sort a hasonló első hálózati csatolóhoz tartozó sor után írjuk be (egyetlen sorba gépelve):

```
/sbin/ipchains -A forward -j MASQ -s
➥$LOCAL_NETWORK2/$LOCAL_NETMASK2 -d 0.0.0.0/0
```

Ezzel készen is vagyunk. Ha nem hibáztunk, újraindítás után (vagy a megfelelő parancsfájlok végrehajtása után) mindkét hálózat internetelérése, valamint a két hálózat közötti forgalom is működni fog.

Összegzés

Mint láthattuk, egy kisméretű Linux-terjesztés segítségével könnyedén megoldhatjuk helyi hálózatunk internetelérését. Az eszközköltség rendkívül csekély, mivel az útválasztót régi és használt gépalkatrészekből is összeállíthatjuk. A rendszer létrehozása és beállítása nem bonyolult, kevés karbantartást igényel és üzembiztos.

Havránek Ferenc
Automatikamérnökként dolgozik. Kedvtelései közé tartozik mindenféle kétkerekű járművön (kerékpár és motor) való közlekedés. Ezenkívül szívesen tölti idejét programozással, nemcsak PC-s, hanem egyéb környezetben is, például mikrovezérlő programokat ír.