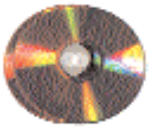
Postfix-csemegék (3. rész)



A Postfix alapvető beállításai után kóstoljunk bele az ínyencségekbe is.

Tekintsük úgy, hogy a tévében jó kis sorozatok epizódjait nézzük – ugyanis mire minden lehetőséget végigpróbálunk, csaknem annyi idő telik el, mintha egy nagyot tévéztünk volna.

Első epizódunk: a webkiszolgálók

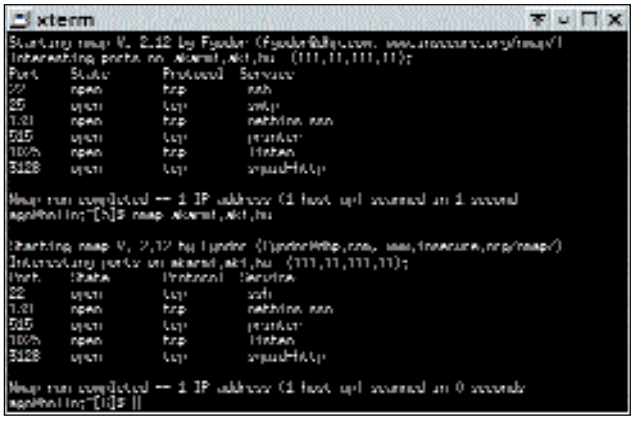
Tételezzük fel, hogy adott egy webkiszolgáló. Ha egy kiszolgálón csak webkiszolgálónak kellene futnia, semmi keresnivalója nem lenne ott egy levelezőkiszolgálónak. De vajon miként kerülnek ki a levelek? Az olyan nem moduláris felépítésű kiszolgálókkal, mint a Sendmail vagy az Exim, biztosan érdekes feladat ezt megoldani, a Postfix azonban képes arra, hogy a különböző levelezési szolgáltatások egymás nélkül is kitűnően tudjanak működni. A mi célunk az, hogy csak a webkiszolgáló vagy a naplófigyelő programok – például a Swatch (lásd még Linuxvilág, 2001. augusztus, 46. oldal) – tudjanak jelentést küldeni, azaz csak tőlük fogadjunk leveleket. Ehhez szükséges lépésként rá kell beszélni a Postfixet, hogy a 25-ös kapun (a kiszolgálók ezen a kapun keresztül kapják a leveleket) csak a 127.0.0.1-es címről fogadjon el kapcsolatot. Ehhez a /etc/postfix/main.cf fájlban keressük meg az `inet_interfaces` kezdetű sort, majd változtassuk meg az értékét (ez alapbeállításban `all`) `localhost` vagy `127.0.0.1` értékre:

```
inet_interfacels = 127.0.0.1
```

Milyen előnnyel bír ez más megoldásokhoz képest? Ha a kiszolgáló levélfogadását csomagszűrő tűzfallal akadályozzuk meg, az pásztázó programmal könnyen felderíthető. Látszik, hogy a kapu nyitott és a kapcsolat szűrt. A mi célunk azonban az, hogy tudomást se szerezzenek róla. Ha választható megoldásként a kiszolgálót más kapun (porton) indítjuk, akkor azt az összes kaput átvizsgáló pásztázás ugyancsak könnyen felderítheti. Nézzük meg azonban a különbséget az *1. képen*, ha a Postfix szolgáltatását használjuk: a felderített kapuk között a levélfogadás nem jelenik meg.

Második epizódunk: SMTP démon nélkül

Az előző részben leírt viselkedésmódot tovább fejleszthetjük. Egyáltalán nem fogadunk levelet az `smtpd`-n (a Postfix 25-ös kapuján figyelő démonon) keresztül, csak a helyi, fájlba írt leveleket továbbítjuk. A leveleket írhatja ember, de program is. Miért lehet erre szükség? Tegyük fel, hogy teljesen kényszeresek vagyunk (bár egy barátom szerint olyan nem létezik, hogy valaki túlságosan paranoid). Félünk attól, hogy valaki mégis megtudja, hogy a gépen levelezőkiszolgáló van, és a csomagszűrő tűzfalat sem mi állítottuk be, hanem egy nem túl hozzáértő „szaki". Számítógépünk hálózati kapcsolón (switch) vagy jelelosztón (hub) osztozik egy vagy több olyan számítógéppel, amelyet feltörtek, mondjuk a legutóbbi `bind` hiba miatt. Mivel a gép rendszergazdája a csomagszűrést rosszul állította be, a hálózati kártya felől bejövő kapcsolatokat is elfogadja a 127.0.0.1-es forráscímről (ez azonban valós körülmények között lehetetlen). Az egyetlen lehetőség ekkor az, ha kikapcsoljuk az `smtpd`-t. Könnyen megtehetjük, csak ismernünk kell hozzá a Postfix szerkezetét. Az összes levelezéssel kapcsolatos démont a `master` program felügyeli. Ebből pedig az következik, hogy amiről nem tud, az nem is működik. A megoldás így roppant egyszerű: a /etc/postfix/master.cf fájlban keressük meg az `smtpd`-re vonatkozó sort, és érvénytelenítsük vagy egyszerűen töröljük ki.
Egy módosított master.cf fájl tartalma látható az *1. listán*.



1. lista Postfix **`smtpd`** démon nélkül

```
#smtp      inet  n   -   -   -   -   smtpd
pickup     fifo  n   n   -   60  1   pickup
cleanup    unix  -   -   -   -   0   cleanup
qmgr       fifo  n   -   -   300 1   qmgr
rewrite    unix  -   -   -   -   -   trivial-rewrite
bounce     unix  -   -   -   -   0   bounce
defer      unix  -   -   -   -   0   bounce
smtp       unix  -   -   -   -   -   smtp
showq      unix  n   -   -   -   -   showq
error      unix  -   -   -   -   -   error
local      unix  -   n   n   -   -   local
```

A vastagon szedett rész mutatja, hogyan kell az adott sort megjegyzéssé tenni.

Harmadik epizódunk: a „nullügyfél"

Felépítése hasonló az előzőéhez, azzal a különbséggel, hogy míg ott lehetőség nyílik a helyi levéltovábbításra, addig itt erre nincsen mód, egyedül az SMTP protokollon keresztüli küldés engedélyezett. Összefoglalva: a nullügyfél olyan számítógép, amely a leveleket csak küldeni tudja. Alkalmazása azokon a kiszolgálókon célravezető, amelyeken a levelezés kimerül a levélnek egy másik kiszolgálóra történő elküldésében. Az első és jelen beállítás közti eltérés, hogy az előbbinél a rendszergazda helyileg is kaphat levelet, például a webkiszolgálótól, az utóbbinál viszont nem. A nullügyfélhez szükséges, hogy Postfixünk tudja, melyik másik kiszolgálót használhatja a levélküldéshez. Ezenkívül minden olyan levelet, amely a gépről „származik", úgy kell elküldenünk, hogy ne látszódjon a gép teljes neve (Full Qualified Domain Name). Az utolsó szükséges beállítás a helyi levéltovábbítás tiltása az `smtpd` mellett. A master.cf és a main.cf beállításait a *2. listán* láthatjuk.

Negyedik epizódunk: a kapcsoltvonali előfizető

Gyakori eset, hogy csak telefonon keresztüli internetkapcsolattal rendelkezünk. Ekkor két út közül választhatunk: az első, hogy olyan

2. lista A Postfix beállítása a nullügyfélhez

```
/etc/postfix/main.cf

# az elküldött leveleknél mindig a tartománynév
# fog látszani a @ jel után
# myorigin = $mydomain a tartomány levelezésért
# felelős számítógépét keressük meg, és rajta
# keresztül küldjük el a levelet
relayhost = $ mydomain

/etc/postfix/master.cf
#smtp     inet  n       -       -       -       -       smtpd
pickup    fifo  n       n       -       60      1       pickup
cleanup   unix  -       -       -       -       0       cleanup
qmgr      fifo  n       -       -       300     1       qmgr
rewrite   unix  -       -       -       -       -       trivial-rewrite
bounce    unix  -       -       -       -       0       bounce
defer     unix  -       -       -       -       0       bounce
smtp      unix  -       -       -       -       -       smtp
showq     unix  n       -       -       -       -       showq
error     unix  -       -       -       -       -       error
#local    unix  -       n       n       -       -       local
```

A vastagon szedett rész mutatja, hogyan kell az adott sort megjegyzéssé tenni.

levelezőprogramot használunk, mely addig gyűjti a leveleket, amíg azt nem mondjuk neki, hogy elküldheti őket. Ennek a megoldásnak előnye, hogy Postfixünket nem kell beállítani – de akkor mire jó? A kérdés megválaszolásához ki kell emelnünk a program hátrányát. Mi történik akkor, ha elfelejtjük elküldeni a leveleket, és akad köztük nagyon fontos is? Nem biztos, hogy újra tudunk csatlakozni, ha „csak” egyetlen ingyenes internetkapcsolattal (például Freestart, Kiwi) bírunk. Ha azonban helyi kiszolgálót használunk, nincs más dolgunk, csak a rendszer tudtára adni, hogy minden kapcsolatfelépítéskor küldje el a leveleket. A kapcsoltvonali beállításhoz az alábbiakat szükséges tudnunk: ki továbbítja majd a levelet; valamint hogy minden levél, amelyet az Internetre továbbítanánk, bent maradjon-e a sorban (queue). Fontos, hogy a kapcsolatot addig ne bontsuk le, amíg minden levelet legalább egyszer meg nem próbáltunk elküldeni. Ezt egy parancsfájlból fogjuk ellenőrizni. A szükséges Postfix-beállítások magyarázattal a következők:

```
/etc/postfix/main.cf
#az alábbi gépen keresztül küldjük el a
#leveleket, ez lehet például a szolgáltató
# smtp-átjárója
relayhost = smtp.szolgaltato.hu
#visszatartjuk az smtp-n keresztül küldendő
#leveleket, amíg nincs kapcsolat
defer_transports = smtp
```

A kimenő sor önműködő kiürítésére parancsfájlt hívunk meg abból a fájlból, amely a vonalas (ppp) kapcsolat felépülése után hajtódik végre. A Debian-rendszer alatt a végrehajtódó parancsállományokat a /etc/ppp/ip-up.d könyvtárban külön fájlokban helyezzük el.
A parancsfájl neve legyen connect. A connect fájl tartalma a következőképpen nézzen ki (a Postfix GyK alapján, de azt módosítva):

```
#!/bin/sh
#kiürítjük a sort
postfix flush

# várunk, hogy minden levél közvetítése
# elkezdődjön, azaz egy smtp-kapcsolat
# elinduljon a távoli kiszolgálók felé,
# várunk 30 másodpercet
sleep 30

# Egészen addig fenntartjuk a kapcsolatot,
# ameddig minden levelet legalább egyszer
# megpróbáltunk elküldeni, ennek keretében
# nem engedélyezzük a poff parancs
# küldését, mentjük más néven, majd
# később visszamásoljuk
mv /usr/bin/poff /usr/bin/poff.off

# amíg új levél tartózkodik a sorban,
# addig várunk, mindig 10 másodperccel
# növeljük a kapcsolat idejét
while mailq | grep '^[^ ]*\*' >/dev/null
       do
        sleep 10
       done
# ha már nincs olyan levél, ami ne
# próbált volna kimenni, akkor
# visszamásoljuk a poff parancsot és
# lezárjuk a kapcsolatot.
mv /usr/bin/poff.off /usr/bin/poff
➥/usr/bin/poff
```

Egyszer nekem szegezték a kérdést, hogy mi történik az alábbi két esetben:

1. Egy rosszindulatú felhasználó olyan parancsfájlt ír, amely a hálózati kapcsolat meglétét ellenőrzi (például egy `ping` parancs kimenetét vizsgálva), majd ha az él, a leveleket iszonyatos mennyiségben kezdi küldeni, mindehhez egy Perl-modult használva.
2. Hálózatba vagyunk kötve, számítógépünk a hálózati és levelező átjáró (smart host). Valaki rossz szándéktól vezérelve kapcsolat idején levelekkel kezd minket bombázni.

Úgy tűnik, ilyenkor sosem tudjuk lekapcsolni a vonalat, és a nappali telefonálás nem olcsó mulatság. Megnyugtatok mindenkit, ez nem fordulhat elő. Miért? Emlékezzünk vissza, hogy a Postfix beállítófájljába beillesztettünk egy olyan sort, amely minden SMTP-kapcsolatot addig késleltet, amíg külön parancsot nem adunk a sor kiürítésére. Ilyen parancsot pedig csak rendszergazdai jogosultságú felhasználó vagy az ő jogaival futó parancsfájl képes végrehajtani. (Ha rajtunk kívül más is rendelkezik rendszergazdai jogosultsággal, akkor már mindegy.) Ellenérv lehet, hogy ha hálózaton nem is, de az első módszert alkalmazva helyileg küldhetünk levelet, hiszen nem kell SMTP-kapcsolatot használni ahhoz, hogy a Postfix elfogadja a levelet. Ez igaz is, használhatjuk a maildrop könyvtárat, de ha nem helyi felhasználónak küldjük a levelet, akkor SMTP-kapcsolaton keresztül kell távoznia, ez pedig késleltetett (deferred) üzemmódban működik. A Postfix modularitása – mint számos más esetben – itt is kisegít bennünket.

Ötödik epizódunk: Postfix és MySQL

Előfordulhat, hogy nem szívesen veszünk fel felhasználókat a rendszerbe, még úgy sem, hogy nem létező héjprogramjuk (shell) van, például a /bin/false vagy a /dev/null. Az is megeshet, hogy már túl sok – például 10 000 – felhasználónk van, ezért a kiszolgáló működése nem hatékony, rendszere esetleg már nem is engedélyezi

újabb felhasználók felvitelét.
Ez esetben eszes megoldásra van szükségünk. Ilyen lehet egy SQL- vagy LDAP-kiszolgáló rendszerbe olvasztása. Ekkor nemcsak a felhasználók meglétének ellenőrzését végezhetjük el, de lehetőséget biztosíthatunk a vezetőség számára, hogy letiltsa a nem fizető ügyfelek azonosítóit, vagy külön kvótákat alkalmazhatunk a különböző előfizetői kategóriák felhasználói számára. A lehetőségek száma szinte a végtelenhez közelít: nem lesz olyan feladat a levelezésben, ahol ne tudnánk hasznosítani a központi adatbázist (többszörözés, adatok ki- és visszamentése stb). A Debian következő, Woody nevű terjesztésében a támogatás már a Postfix-csomagba lesz építve. Addig azonban saját magunknak kell fordítanunk a csomagot, ha azt akarjuk, hogy MySQL-támogatás is legyen. Ehhez az include és header fájlok miatt szükségünk lesz a MySQL fejlesztői csomagjára. Ha a MySQL-t is forrásból fordítjuk, ezek is felkerülnek a telepítés után. A Postfix fordításával már foglalkoztunk a legelső részben. Mivel még mindig nincs configure parancsállomány, a make programnak kell átadni változóként, de csak akkor, amennyiben a programba a MySQL táblaalapú adattárolását is be akarjuk építeni. Ha nem ez az első fordítás, mindenekelőtt takarítsunk egy kicsit:

```
# make tidy
```

Ezt követően adjuk ki a következő parancsot:

```
# make -f Makefile.init makefiles
➥'CCARGS=-DHAS_MYSQL -I/usr/local/include/mysql'
➥'AUXLIBS=/usr/local/lib/mysql/libmysqlclient.a -lm'
```

Ha a Makefile elkészült a fenti beállításokkal, kiadhatjuk a parancsot:

```
# make
```

A szokásos folyamat zajlik le, mindegyik démon külön-külön lefordítódik, és elindíthatjuk a telepítő és beállító parancsfájlt:

```
# sh INSTALL.sh
```

Ha elindítjuk a Postfixet, a postconf programot használva láthatjuk, hogy a használható adattároló-típusok között megjelent a MySQL-támogatás:

```
# postconf -m
nis
regexp
environ
mysql
btree
unix
hash
```

Amennyiben újabb kiadású MySQL-kiszolgálót használunk és csomagból telepítettünk, rendelkeznünk kell a zlib fejlesztői csomagjával (zlib-dev), ugyanis most már a libmysqlclient.a tárgykönyvtárat a zlib segítségével készítik. Ha viszont a MySQL-t forrásból telepítettük, a fordítást megelőző beállítóprogram alapértelmezettként a zlib-csomagokat használja. Ekkor az SQL-támogatás beépítéséhez a Makefile-t a következőképpen kell indítanunk:

```
# make -f Makefile.init makefiles
➥'CCARGS=-DHAS_MYSQL -I/usr/local/include/mysql'
➥'AUXLIBS=/usr/local/lib/mysql/
➥libmysqlclient.a -lm -lz'
```

Látszik tehát, hogy az egész mindössze egy `-lz` lehetőséggel bővült. Csak három karakter, de ha nem tudjuk, hogy oda kell rakni, sokat szenvedhetünk, mire rájövünk a hiba okára. A jelenlegi Postfix-terjesztésből hiányzik e tény megemlítése, mégis érdemes megnézni a forráscsomagban lévő MYSQL_README fájlt, mert ez a hivatalos leírás, és amennyiben később változtatnak valamin, ott biztosan megtaláljuk ezeket. A csomagban szintén találhatunk egy leírást arról, hogy milyen táblákat kell létrehozni az adatbázis-kiszolgálóban, és milyen formátumban kell kitölteni egy SQL-t használó mapfájlt. Ha ezt követően ilyen fájlra hivatkozunk, a típusnál a `mysql`-t kell megadni, ilyen formában:

```
alias_maps = mysql:/etc/postfix/mysql-alias.cf
```

Ha ezt a típust szeretnénk a hash helyett alapértelmezetté tenni, akkor a

```
default_database_type = hash
```

sort a következőre kell cserélni:

```
default_database_type = mysql
```

Éjféli híradó: helyi levéltovábbítás

A Postfix helyi levéltovábbításra saját helyi programját használja, de bármikor eltérhetünk tőle, erre ad lehetőséget a mailbox_command érték. Amíg ennek értéke üres, addig a helyi démon fog végrehajtódni, eltérni csak értékadással lehet. Ha a procmailt akarjuk levéltovábbításra használni, az értéket a main.cf-ben állítsuk be rá:

```
mailbox_command = /usr/bin/procmail
```

Figyelem! Postfix használata esetén a `procmail`-nek set-uid bittel kell rendelkeznie! Ez teszi lehetővé, hogy mindig a címzett felhasználói jogaival tudjon futni és beleírjon a felhasználó postafiókjába. Vigyázat! Ha nem `mbox`, hanem `Maldir/` formátumban (minden levél külön fájlban) tároljuk a leveleinket, a `procmail` nem alapbeállításaival fog működni. `Maildir`-be mentéshez használjuk inkább a helyi démont, ami el is készíti a szükséges könyvtárakat, amennyiben még nem léteznek.

A fogmosás: ha elsőre nem sikerül…

Mi történik akkor, ha a rendszeren kétféle felhasználó létezik? Egy, aki benne van a /etc/passwd fájlban, és egy, aki viszont nem, mert mondjuk a Cyrus-IMAP saját virtuális felhasználói között található. Ekkor nem a `mailbox_transport`, hanem a `fallback_transport` értéket kell beállítani. Ha a master.cf fájlban létezik például a `cyrus` bejegyzés (amennyiben nem töröltük ki, alapértelmezésben megtalálható benne), megtehetjük, hogy először megpróbáljuk a /etc/passwd-ben megtalálni a felhasználót, majd ha ott nincs, akkor a Cyrusnak átdobni a levelet. Az ehhez szükséges beállítások a /etc/postfix/main.cf-ben találhatók:

```
mailbox_command =
fallback_transport = cyrus
```

Remélem, mindenkinek sikerült megtalálnia az őt legjobban érdeklő témát. A következő alkalommal néhány nem szokványos beállítást, valamint egy irodai rendszer beüzemelését mutatjuk be.

Deim Ágoston (ago@lsc.hu)
Kedveli a sört, szereti a futást és imádja Szabó Lőrinc verseit. Nem hisz vakon egyik rendszerben sem. Vonzódik a BSD-hez is. Tagja az LME-nek és a MBE-nek. Mottója: a gép nem lehet fontosabb az embernél.