Teljes értékű levelezőrendszer

Építsünk több tartományt kezelő SMTP-levélkiszolgálót egyetlen gépen.

Bár e cikket útmutatónak szántuk, amelynek alapján a Postfix, az OpenLDAP és a Courier-IMAP segítségével felépíthetjük a saját teljes levelezőkiszolgáló rendszerünket, azzal nem foglalkozunk, hogyan választottuk ki épp ezeket az összetevőket, hiszen ennek kifejtése önmagában megérne egy cikket. Célunk, hogy egyetlen gépen állítsunk fel egy több tartományt kezelni képes SMTP-levélkiszolgálót, távoli IMAP-eléréssel.

Azt szeretnénk, ha nemcsak a héjprogram-azonosítóval rendelkező emberek számára kézbesítenének leveleket, hanem a héjprogram-azonosítóval nem rendelkező embereknek is lehetne IMAP-azonosítójuk. Így az azonosítókat két osztályba soroljuk: helyi és virtuális osztályba. A helyi azonosítók azok, amelyeknek van parancsértelmező elérésük. Ők a saját felhasználónevüket és jelszavukat használhatják az IMAP elérésére. A virtuális azonosítókhoz olyan felhasználónevet és jelszavat rendelünk, amely csak az IMAP-bejelentkezéshez használható. A cikk további részében a helyi és a virtuális fogalmat ilyen értelemben fogjuk alkalmazni.

Áttekintés

Az *1. ábra* bemutatja, hogyan kapcsolódik egymáshoz a Postfix, a Courier, a Procmail és az OpenLDAP. A helyi azonosítók a */etc/password* fájlban találhatók, az azonosítást pedig a betölthető azonosítómodulok (Pluggable Authentication Module, azaz a PAM) végzik. A virtuális azonosítók adatait az LDAP könyvtárban tároljuk. Az LDAP az azonosítókeresési és -hitelesítési lehetőségeket egyaránt támogatja. Ha szükséges, az LDAP könyvtárat ki lehet hagyni, így azonban jóval nehezebb lesz karbantartani a virtuális azonosítók adatait. Megfelelő beállításfájlokkal mind a Postfix, mind a Courier támogatja a virtuális azonosítókat, ezek azonban eltérő formátumúak.

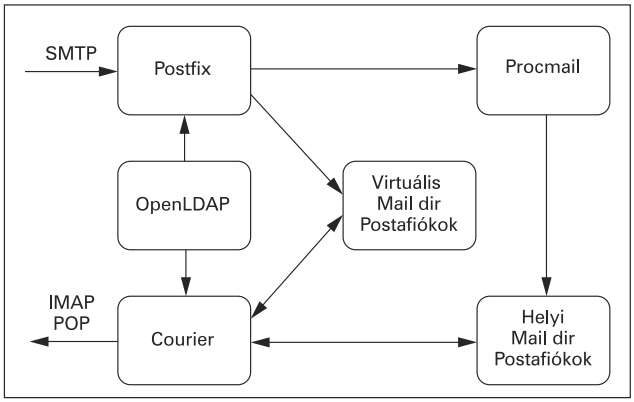
Az SMTP-ről érkező leveleket a Postfix fogadja. Az ismeretlen (helyi vagy virtuális) azonosítóról érkező leveleket elutasítja. A virtuális azonosítókhoz maga juttatja el a levelet, a helyi felhasználókhoz pedig a Procmailt használja MDA-ként.

A Courier az IMAP és a POP protokollokon keresztül távoli elérést nyújt levélládákhoz.

A levélláda helye

A helyi azonosítók leveleit Maildir formátumban a saját könyvtárukban tároljuk, a `${HOME}/Maildir/` alkönyvtár alatt. Általánosan alkalmazott megoldás, hogy a Maildir kézbesítését a */var/spool/mail* helyett az azonosító saját könyvtárába végezzük. Mind a Postfix, mind a Courier tökéletesen működik ezzel a szabványos módszerrel.

A helyi azonosítókkal ellentétben a virtuális azonosítók leveleihez nem tartozik szabványos hely. Ezért külön Unix-azonosítót készítettünk `vmail` néven, ahol az összes virtuális felhasználó leveleit elhelyezhetjük. Minden virtuális tartományhoz tartozik egy alkönyvtár a *~vmail/domains/* könyvtárban. Például ha van egy <john@pelda.hu> azonosítónk, a levelek a *~vmail/domains/pelda.hu/john/* könyvtárba kerülnek Maildir formátumban.



1. ábra Általános kiépítés

Az LDAP könyvtár megtervezése

Könyvtárunkat számtalan módon megtervezhetjük, a témát most nem elemezzük az összes lehetséges szempont szerint. Cikkünkben feltételezzük az LDAP-fogalmak és szaknyelv általános ismeretét.

A faszerkezet

Gyökérutótagként a cég tartománynevét (***myhosting.example***) választottuk. A Postfix és a Courier egyaránt az `o=hosting, dc=myhosting, dc=example` alfában keresi majd az elektronikus levél adatait. Az `o=accounts, dc=myhosting, dc=example` alfa bemutatja, hogyan tudnánk egyetlen könyvtárba beilleszteni a héjprogram-azonosítók PAM-adatait is, ez azonban a levelezéshez most nem szükséges. Minden kezelt tartomány saját szervezettel rendelkezik a hosting szervezet alatt. Minden elektronikus levélazonosító a tartományok alfájába kerül. Ennek megfelelően a <user2@domain2.example> elektronikus levélcím megkülönböztető neve:

```
mail=user2@domain2.example,o=domain2.example,
 o=hosting,dc=myhosting,dc=example
```

A fenti tervezet elég megbízható, hiszen az azonosítókat soha nem visszük át másik tartomány alá. Tervünk emellett rugalmas is, hiszen az egyes tartományfákat – amennyiben szükséges – tetszés átszabhatjuk. Minden tartományhoz tartozik egy postamester-bejegyzés, ami kettős feladatot lát el. Elsődleges célja az elérési jogosultságok szabályozása, emellett levéltovábbító elektronikus levélcímként is üzemel. Minden tartományhoz tartoznia kell egy kitiltandók listájának (abuse alias), amely a rendszergazdának továbbítja a leveleket.

Sémaválasztás

A séma mutatja meg (objektumosztályok megadásával), hogy milyen tulajdonságai (attributum) lehetnek egy bejegyzésnek. Az OpenLDAP rendszerhez szállított sémák közül egyik sem felel meg igazán olyan bejegyzésekhez, amelyeket kizárólag

1. tábla **A courierMailAccount**

Attributum	Kötelező	Leírás
mail	Igen	A teljes levélcím
homeDirectory	Igen	Az alapkönyvtár, ahol a leveleket tároljuk
uidNumber	Igen	Az üzenetek tárolására használt azonosítóhoz tartozó felhasználói ID
gidNumber	Igen	Az üzenetek tárolására használt azonosítóhoz tartozó csoport ID

2. tábla A courierMailAlias

Attributum	Szükséges	Leírás
mail	Igen	Teljes levélcím
maildrop	Igen	Levélcím, ahová továbbítani kell. Lehet helyi álnév, vagy távoli levélcím

elektronikus levelesládákhoz és levéltovábbításhoz szeretnénk használni. Ezért inkább a Courier-terjesztésben található sémát használjuk fel. Érdemes megnézni a qmail-LDAP Projekttel érkező sémát is.

A Courier-séma

A virtuálislevél-azonosítókhoz használt `courierMailAccount` objektumosztályt az *1. táblázatban* foglaltuk össze. A *2. táblázatban* látható `courierMailAlias` objektumosztályt azokhoz az elektronikus levélcímekhez használjuk, amelyek más címekre továbbítanak neveket.

A `courierMailAccount` objektumosztály sajnos nem felel meg tökéletesen a céljainknak. Az `uidNumber` és `gidNumber` számunkra felesleges, hiszen minden levelet a `vmail` azonosítóra küldünk. Valamilyen álértéket mégis be kell írnunk, mivel a séma megköveteli a jelenlétét. Figyeljük meg, hogy ha több Unix-azonosító közt osztottuk volna szét a virtuális azonosítókat, ezeknek az értékeknek is lenne értelmük. Szükségünk lesz a levélláda- (mailbox) tulajdonságra, mivel ez alapján tudjuk megállapítani a levélláda helyét a fájlrendszeren. A levélláda bejegyzésnek perjellel kell végződnie, így jelezve, hogy Maildir stílusú levelesládáról van szó. A `userPassword` kapcsolóra úgyszintén szükségünk lesz, hiszen az elektronikus levélazonosítókhoz jelszavakat kell rendelnünk, hogy IMAP- vagy POP-rendszeren keresztül elérhetők legyenek. A többi elhagyható tulajdonságot nem használjuk.

A `courierMailAlias` objektumosztály éppen megfelel nekünk. Csak a két kötelező kapcsolót fogjuk használni, az elhagyhatók közül egyikkel sem foglalkozunk. A `maildrop` tulajdonság egy másik elektronikus levélcím vagy a helyi gép egy azonosítója lehet.

Jogosultságszabályozás

Az OpenLDAP több lehetőséget is kínál a jogosultság szabályozásra. Alapértelmezés szerint a rendszergazdai azonosítónak a fa összes bejegyzésére írási és olvasási joga van. A felügyeleti feladatok egy részét érdemes kezelendő tartományonként külön azonosítókra ruházni, hogy a kisebb változtatásokat a rendszergazdai azonosító elérése nélkül is elvégezhessék. Ezt úgy érhetjük el, hogy azokban a bejegyzésekben, ahol felügyeleti előjogokat akarunk adni, a ***postmaster*** (postamester) bejegyzést `organizationalRole`-á tesszük a `roleOccupant` tulajdonsággal. Aztán az OpenLDAP-ot beállíthatjuk úgy is, hogy csak e csoport tagjai érhessék el.

Megvalósítás

Ebben a részben bemutatjuk, hogyan valósíthatunk meg egy virtuális levelezést. Minden apró részletet nem fogunk ismertetni, csak azokra térünk ki, amelyek a szabványos telepítésen felül szükségesek.

Alább felsoroltuk azokat a programokat (és változatszámokat), amelyeken az alkalmazást kipróbáltuk:

- Red Hat Linux 6.2, 7.1 vagy 7.2;
- Postfix 1.1.x;
- OpenLDAP 2.0.21;
- Courier-IMAP 1.4.1;
- Procmail 3.22.

Létre kell hoznunk a `vmail`-azonosítót, majd a *~vmail/domains/* könyvtárat. Továbbá szükségünk lesz még két azonosítóra és két csoportra a Postfixhez – a Postfix telepítési útmutatójának megfelelően.

Az OpenLDAP fordításához és telepítéséhez nem lesz szükségünk különleges ismeretekre, így az utasításokat nézzük meg a leírásban. Éles alkalmazás esetén előbb olvassuk el, hogyan lehet az OpenLDAP rendszert nem rendszergazdaként futtatni a `chroot`-környezet felállításával és másolással. Ebben a cikkben azt mutatjuk meg, hogyan kell a `slapd`-t egyetlen kiszolgálón beállítani, létrehozni az alapfaszerkezetet, illetve beszúrni néhány alapadatot az LDAP-könyvtárba.

A slapd beállítása

Szükségünk lesz a Courier sémafájlra, ezért a Courier-terjesztés ***authlib/authldap.schema*** állományát másoljuk a */usr/local/etc/openldap/schema/courier.schema* helyre. A Courier-séma a ***cosine.schema*** és a ***nis.schema*** sémáktól függ. Adjuk a következő sorokat a ***slapd.conf*** fájlhoz:

```
include
/usr/local/etc/openldap/schema/cosine.schema
include
/usr/local/etc/openldap/schema/nis.schema
include
/usr/local/etc/openldap/schema/courier.schema
```

Ezután az adatbázis-meghatározásokat a ***slapd.conf*** fájlban a következő bejegyzésekkel állítsuk be:

```
directory       /usr/local/var/openldap-ldbm
database        ldbm
suffix          "dc=myhosting,dc=example"
```

A `database` kulcsszó meghatározza, hogy milyen háttértárat használunk (itt LDBM adatbázist adtunk meg). A `directory` kulcsszó az LDBM adatbázis elérési útját adja meg. Ne feledjük, hogy az itt megadott elérési útnak a `slapd` indítása előtt már léteznie kell, és a `slapd` írási és olvasási jogokkal kell rendelkezzen a könyvtáron (és természetesen az sem árt, ha execute jogosultsága is van, különben nem fog menni – a ford.). A `suffix` kulcsszó az adatbázishoz rendelt `root` utótagot adja meg. A következő néhány sor a szuperfelhasználói, más néven `root` azonosítót adja meg:

```
rootdn
"cn=Manager,dc=myhosting,dc=example"
```

```
rootpw
{SSHA}ra0sD47QP32ASAlaAhF8kgi+8Aflbgr7
```

A `rootdn` bejegyzésnek korlátlan elérési jogai vannak a teljes adatbázis felett, ezért jelszavát a tényleges adatbázison kívül őrizzük. A `rootpw` kulcsszóval megadott jelszót mindig kódolt formában tároljuk. Soha ne írjunk be egyszerű szöveget jelszónak. A szöveges jelszó (például: secret) titkosított jelszóvá kódolását a `slappasswd` paranccsal végezzük el:

```
% slappasswd
New password: secret
Re-enter new password: secret
{SSHA}ra0sD47QP32ASAlaAhF8kgi+8Aflbgr7
```

A kiírt sort vegyük ki a `slappasswd`-ből és másoljuk a ***slapd.conf*** fájlba, ahogy a fenti példában is tettük.
A keresések gyorsítása érdekében az általánosan használt tulajdonságokhoz indexeket készíthetünk:

```
index    objectClass   pres,eq
index    mail,cn        eq,sub
```

A ***slapd.conf*** utolsó része a jogosultságszabályozás.

A könyvtárfa létrehozása

A `slapd` beállítása után ideje hozzáfognunk az LDAP könyvtár feltöltéséhez. Az OpenLDAP-pal szállított parancssoros eszközöket fogjuk használni, és LDIF fájlokat hozunk létre a könyvtár módosításához.
Az első lépés a `root` csomóponthoz tartozó alapszerkezet elkészítése: ez az otthont adó szervezet (hosting organization) és a `rootdn`-hez tartozó bejegyzés. Hozzunk létre egy fájlt ***base.ldif*** néven a következő tartalommal:

```
dn: dc=myhosting, dc=example
 objectClass: top

 dn: cn=Manager, dc=myhosting, dc=example
 objectClass: top
 objectClass: organizationalRole
 cn: Manager

 dn: o=hosting, dc=myhosting, dc=example
 objectClass: top
 objectClass: organization
 o: hosting
```

Használjuk az `ldapadd` parancsot a rendszergazdai azonosítót használva, hogy bevigyük a fenti LDIF-et:

```
ldapadd -x -D
"cn=Manager,dc=myhosting,dc=example" \
-w secret -f base.ldif
```

Tartomány felvitele

Immár felvihetjük a tartományokat a „hosting” fa alá. Minden tartománynak rendelkeznie kell legalább a postamesterrel és a kitiltandók listája bejegyzésekkel (abuse entries).
A ***domain1.example*** fa létrehozásához készítsünk egy ***domain1.example.ldif*** nevű fájlt a következő tartalommal:

```
dn: o=domain1.example, o=hosting,
dc=myhosting,
 dc=example
objectClass: top
objectClass: organization
o: domain1.example

dn: cn=postmaster, o=domain1.example,
o=hosting,
 dc=myhosting, dc=example
objectClass: top
objectClass: organizationalRole
objectClass: CourierMailAlias
cn: postmaster
mail: postmaster@domain1.example
maildrop: postmaster

dn: mail=abuse@domain1.example,
o=domain1.example,
 o=hosting, dc=myhosting, dc=example
objectClass: top
objectClass: CourierMailAlias
mail: abuse@domain1.example
maildrop: abuse
```

Figyeljük meg, hogy a `maildrop` kapcsolók mindig helyi elektronikus levélazonosítók, és a postamesterhez, illetve a */etc/**aliases* fájlban megadott álnevekre továbbítódnak. A ***postmaster*** szabályban nem adtunk meg azonosítót, így jelenleg kizárólag a rendszergazdai azonosítón keresztül lehet új azonosítókat létrehozni. Vigyük fel a tartományt a következő paranccsal:

```
ldapadd -x -D
"cn=Manager,dc=myhosting,dc=example" \
➥-w secret  -f domain1.example.ldif
```

Azonosítók felvitele

Vigyük fel a <user1@domain1.example> levélcímmel rendelkező felhasználót. Egyúttal ennek a felhasználónak adjunk postamester-előjogokat a ***domain1.example*** tartományra. Készítsük el a ***user1.domain1.example.ldif*** fájlt a következő tartalommal:

```
dn: mail=user1@domain1.example,
o=domain1.example,
 o=hosting, dc=myhosting, dc=example
objectClass: top
objectClass: CourierMailAccount
mail: user1@domain1.example
homeDirectory: /home/vmail/domains
uidNumber: 101
gidNumber: 101
mailbox: domain1.example/user1

dn: cn=postmaster, o=domain1.example,
o=hosting,
dc=myhosting, dc=example
changetype: modify
add: roleOccupant
roleOccupant: mail=user1@domain1.example,
 o=domain1.example, o=hosting,
 dc=myhosting, dc=example
```

Az első rész az azonosítóhoz tartozó bejegyzést viszi be.
A `home directory` és a `mailbox` a fájlrendszeren fizikailag

elérhető levelesládára mutat. Az `uidNumber` és `gidNumber` kapcsolók kötelezően megadandók, de mivel mi nem használjuk őket, a 101-es próbaértékkel (dummy value) töltöttük fel. A második rész módosítja a postmaster bejegyzést – hozzáadja a `roleOccupant` kapcsolót a `user1@domain1.example` DN-t használva. Hozzuk létre ezt a bejegyzést:

```
ldapadd -x -D
"cn=Manager,dc=myhosting,dc=example"
➥-w secret -f user1.domain1.example.ldif
```

Mivel az azonosítóhoz még nem tartozik jelszó, hiába rendelkezik postamesteri jogosultságokkal, nem tudjuk hitelesíteni. Az `ldappasswd` paranccsal állítsuk be a `user1` kezdeti jelszavát:

```
ldappasswd -x -D "$DN" -w $PW -s user1
➥"mail=user1@domain1.example,
o=domain1.example,
o=hosting, dc=myhosting, dc=example"
```

A további tartományokat és azonosítókat hasonló LDIF fájlokkal vihetjük fel. Az LDIF fájlok felvitele kézi módszerrel meglehetősen fárasztó, ráadásul könnyen hibázhatunk is. Később más módszereket is mutatunk a felügyeletre.

Postfix

A Postfix rendszernek csak azokat a részeit tárgyaljuk, amelyek a levélkezelésre vonatkoznak.
Töltsük le a Postfix-forrást és csomagoljuk ki. Újra kell építenünk a Postfix `Makefile`-okat, hogy figyelembe vegyék és hivatkozzanak (link) az LDAP programkönyvtárakra. Hajtsuk végre a következő parancsot:

```
make makefiles CCARGS="-I/usr/local/include
➥-DHAS_LDAP" AUXLIBS="-L/usr/local/lib -lldap
➥-L/usr/local/lib -llber"
```

Innentől követhetjük a szokásos Postfix fordítási és telepítési útmutatásokat, amelyeket az ***INSTALL*** és az ***LDAP_README*** fájlokban találunk.

A Postfix beállítása

Ha az alább bemutatott beállítási példák bármelyikéhez nem neveztünk meg kifejezetten valamilyen fájlt, akkor azt minden bizonnyal a ***main.cf*** fájlban találjuk.
A szállítási tábla (transport table) a tartományokat rendeli az (***/etc/postfix/master.cf*** fájlban megadott) üzenetkézbesítő egységekhez (message delivery transports), illetve a továbbító gazdagépekhez. Mi virtuális tartományainkat a Postfixszel érkező virtuális kézbesítő ügynököknek szeretnénk átadni. A szállítási tábla tehát valahogy így néz ki:

```
domain1.example             virtual:
domain2.example             virtual:
```

Miután egyszerű szövegfájlként elkészítettük a szállítási táblát, a `postmap` utasítással (lásd man postmap) át kell alakítanunk bináris DB állománnyá. Ha ezzel megvagyunk, mutassuk meg a Postfix rendszernek, hogy van egy szállítási táblánk, és hogy azt merre találhatja meg. Azt is tudatnunk kell a Postfix programmal, hogy ezekre a tartományokra leveleket várunk.

Ezt a `transport_maps` és `mydestination` kulcsszavakkal tehetjük meg:

```
transport_maps = hash:/etc/postfix/transport
➥mydestination = $myhostname,
➥localhost.$mydomain,
➥   $mydomain, $transport_maps
```

Könnyen megadhatunk többszörös LDAP-forrásokat is. Az LDAP-forráskapcsolók leírását a ***README_FILES/LDAP_README*** fájlban olvashatjuk, a Postfix forrásában. A kapcsolónevek `<ldapforrÆs>_kapcsol` alakúak. Az LDAP-forrás nevét a használatával adjuk meg. A ***main.cf*** fájlban keresésenként egy-egy LDAP forrásmeghatározásra lesz szükségünk.

Álnevek

Az első LDAP-forrásmeghatározást a virtuális álnevekhez hozzuk létre. Ezt az LDAP-forrást „aliases”-nek, azaz álneveknek neveztük el. Beállításunk szerint az LDAP-kiszolgáló a helyi gépen fut. A keresés alapja az LDAP-kiszolgálónkon megadott „hosting” alfa lesz. Azokat az elemeket kérjük le, ahol a `mail` elem megegyezik a levél címzettjével, illetve amelyek a `courierMailAlias` objektumosztályba tartoznak. Az álnévhez rendelt célszemélyt a `maildrop` tulajdonsága tartalmazza. A Postfix nem fog felhasználóként bejelentkezni, hanem anonim keresést végez:

```
aliases_server_host = localhost
aliases_search_base =
  o=hosting,dc=myhosting,dc=example
aliases_query_filter =
  (&(mail=%s)(objectClass=CourierMailAlias))
aliases_result_attribute = maildrop
aliases_bind = no
```

Azonosítók

Amikor az `accounts` (azonosítók) forrást használjuk, olyan bejegyzéseket keresünk, amelyek a `courierMailAccount` objektumosztályba tartoznak. Eredményként a mailbox tulajdonságot kérjük vissza:

```
accounts_server_host = localhost
accounts_search_base =
  o=hosting,dc=myhosting,dc=example
accounts_query_filter =
  (&(mail=%s)(objectClass=CourierMailAccount))
accounts_result_attribute = mailbox
accounts_bind = no
```

Az azonosítókhoz egy második, `accountsmap` nevű forrást is létre kell hoznunk, hogy az azonosítókat `catchall` (minden elem megvizsgálása) esetén is le tudjuk kérni. Enélkül az álnevekben használt `catchall` felülbírálná a tartományok virtuális azonosítóit:

```
accountsmap_server_host = localhost
accountsmap_search_base =
o=hosting,dc=myhosting,dc=example
accountsmap_query_filter =
(&(mail=%s)(objectClass=CourierMailAccount
accountsmap_result_attribute = mail
accountsmap_bind = no
```

Miután megadtuk az `aliases` és az `accountsmap` LDAP-forrásokat, tudassuk a változtatásokat a Postfix programmal, és állítsuk be a `main.cf` `virtual_maps` kapcsolóját:

```
virtual_maps = ldap:aliases
```

A példa kedvéért tételezzük fel, hogy már készítettünk egy `vmail` nevű Unix-azonosítót, amely a 125-ös UID, és a 120-as GID értékeket birtokolja, a saját könyvtára pedig a */home/vmail*:

```
:virtual_mailbox_base = /home/vmail/domains
virtual_mailbox_maps = ldap:accounts
virtual_minimum_uid = 125
virtual_uid_maps = static:125
virtual_gid_maps = static:120
```

A `virtual_uid_maps` és `virtual_gid_maps` értékeit egy különleges statikus térképhez rendeltük, belekódolva a `vmail` azonosító UID és GID értékeit. Az összes itt bemutatott kapcsoló teljes leírását elolvashatjuk a Postfix forrásával együtt kapott ***README_FILES/VIRTUAL_README*** állományban. Át kell szerkesztenünk a `local_recipient_maps` kapcsolót, hogy a `virtual_mailbox_maps`-ban keresgéljen, így a Postfix tudni fogja, hogy mely azonosítók érvényesek. Ez azért szükséges, mert a Postfix el tudja utasítani az ismeretlen azonosítókra érkező leveleket:

```
local_recipient_maps = $alias_maps
  unix:passwd.byname $virtual_mailbox_maps
```

Courier

Semmilyen különleges utasítást nem kell használnunk a Courier telepítéséhez, úgyhogy útmutatásért forduljunk nyugodtan a leíráshoz. Meg fogja találni és be fogja fordítani az LDAP-ot. Érdemes meggondolni a `--enable-workarounds-for-imap-client-bugs` kapcsoló használatát a `./configure` futtatásakor, mert enélkül a Netscape-felhasználóknak gondjaik támadhatnak, amikor a kiszolgálónkat akarják használni.
A Courier külön azonosítódémont használ, hogy az azonosítást elválassza a rendszer egyéb részeitől. Állítsuk be úgy, hogy az érvényes levélazonosítókat LDAP és PAM alatt is megtalálja. Ezt az ***authdaemonrc*** fájlban az `authmodulelist` kapcsolóval adhatjuk meg:

```
authmodulelist="authldap authpam"
```

Minden LDAP-kapcsoló az `authldaprc`-ben található. A legtöbb kapcsoló magától értetődő. A Courier-séma használatához azonban el kell végeznünk néhány változtatást. Ezenkívül az összes virtuális azonosítót a `vmail` azonosítóhoz kell rendelnünk. Összefoglalóan a következő változtatásokat kell elvégeznünk az `authldaprc`-ben:

```
LDAP_GLOB_UID           vmail
LDAP_GLOB_GID           vmail
LDAP_HOMEDIR            homeDirectory
LDAP_MAILDIR            mailbox
LDAP_CRYPTPW            userPassword
```

Három további beállítás alkalmazását érdemes megfontolnunk, ezek: a `LDAP_AUTHBIND`, a `LDAP_BINDDN` és az `LDAP_BINDPW` – mindhárom a felhasználó azonosításához tartozik. Az `LDAP_AUTHBIND` kulcsszó és a `LDAP_BINDDN`, `LDAP_BINDPW` páros kölcsönösen kizárja egymást. Mi az `LDAP_AUTHBIND` használatát javasoljuk. Az `authldaprc` egyik megjegyzése memóriaszívást említ az OpenLDAP-ban a LDAP_AUTHBIND használatakor, ezt azonban az OpenLDAP 2.0.19 változatában már kijavították. Ha az `LDAP_BINDDN` és `LDAP_BINDPW` kulcsszavakat használjuk, jelszavainkat csak a crypt, MD5 és SHA algoritmusokkal kódolhatjuk. Az SMD5 és az SSHA nem lesz elérhető. Továbbá ha `LDAP_BINDPW` adunk meg, az LDAP jelszó egyszerű szövegként kerül az `authldaprc` fájlba. Az LDAP-jelszavakat egyszerű szövegként tárolni komoly biztonsági rést jelent, így ha csak lehet, inkább az `LDAP_AUTHBIND` módszert használjuk.
Az utolsó változtatás, amit el kell végeznünk, az IMAP-kiszolgáló beindítása az `IMAPDSTART` kapcsoló YES-re történő állításával. Mostantól a `courier-imap.sysvinit` indító parancsfájlt használhatjuk az IMAP-démon indításához és leállításához.

Felügyelet

A legtöbb felügyeleti feladat, az azonosítók és álnevek felvétele, módosítása és törlése az LDAP könyvtár módosítását igényli. Ezt az OpenLDAP parancssoros eszköz segítségével vagy valamilyen általános LDAP böngésző (például a gq) alkalmazásával tehetjük meg. Jelenleg egy Jamm nevű webfelügyeleti alkalmazáson dolgozunk, amely lényegében egy Java és JSP nyelven íródott alkalmazásspecifikus LDAP-böngésző lesz. Saját LDAP-sémával is rendelkezik, amely tulajdonképpen egy kis mértékben módosított Courier-séma. A Jamm jelenleg is használható és folyamatosan fejlődik – a legfrissebb tájékoztatást a Jamm SourceForge honlapon találjuk.

Megjegyzések az azonosítókészítéshez

Amikor azonosítókat és álneveket készítünk, az LDAP adatbázisban azok azonnal működővé válnak – feltételezve, hogy a levelezőrendszer ezen felhasználja őket. A virtuális azonosítók létrehozásakor ne feledjük, hogy a hozzájuk tartozó Unix-könyvtár nem jön létre a `~vmail`-ben. Ezt azonban orvosolhatjuk, mivel a Postfix virtuális kézbesítőügynöke az első levél megérkezésekor létrehozza a szükséges könyvtárakat. Éppen ezért azt javasoljuk, hogy amint létrehoztuk az azonosítót, küldjünk egy üdvözlőlevelet.

Linux Journal 2003. február, 106. szám

Dave Dribin
1991 óta használ Unixot és 1993 óta foglalkozik Linuxszal. 1995 óta hivatásszerűen fejleszt programokat Unix-rendszereken vagy -rendszerekre. Dave jelenleg független tanácsadóként dolgozik a realtorsi National Associationnél.

Keith Garner
1994 januárja óta használja a Linuxot. 1997 óta hivatásszerűen fejleszt Unix-programokat. Keith jelenleg a realtorsi National Association alkalmazásában áll.

KAPCSOLÓDÓ CÍMEK

Courier-IMAP ➲ http://www.inter7.com/courierimap
OpenLDAP ➲ http://www.openldap.org
Postfix ➲ http://www.postfix.org
Procmail ➲ http://www.procmail.org