

KISKÖRÖS VÁROS ÖNKORMÁNYZAT
POLGÁRMESTERI HIVATALA

6200 Kiskőrös, Petőfi tér 1.
Tel.: (06) 78-311-122; Fax: (06) 78-312-160

MAILMAN

The GNU Mailing List Management System

Telepítési, felhasználói és adminisztrátori útmutató

A dokumentáció és az alkalmazás telepítő CD-je az **ÁROP 1.A.2 pályázat megvalósításaként a „**3. A partnerség erősítése**” pontjaihoz kapcsolódóan került leszállításra Kiskőrös Város Önkormányzata részére**

Copyright (C) 1998-2004 by the Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA

A kiadás dátuma: 2009.11.16.



A Projekt az Európai Unió támogatásával,
az Európai Szociális Alap társfinanszírozásával valósul meg.
Készítette: 4 Sales Systems Kft.

BEVEZETÉS

A GNU Mailman egy levelezőlista-kezelő program GNU Public License (GPL) alatt. A projekt nevét, a "Mailman"-t kezdő nagy `M'-el és kis középső `m'-el kell helyesen írni.

A Mailman elsősorban Python-t, egy ingyenes objektum-orientált nyelvet használ. Biztonsági célból néhány helyen ANSI C kódot is tartalmaz.

A Mailman első fejlesztője John Viega volt. Ken Manheimer segítségével (az 1.0b3 verzióig) folytatódott a fejlesztés. A későbbiekben és az 1.0-ás végleges verzióban már több programozó vett részt a fejlesztésben. A vezető programozók közé Barry Warsaw, Ken Manheimer, Scott Cotton, Harald Meland és John Viega tartozott. Az újabb verziók fejlesztését Barry Warsaw vezeti; a fejlesztésben résztvevők egy részének névsorát az ACKNOWLEDGMENTS állományban lehet olvasni. A Mailman 2.0-ban található Pipermail kódjának fejlesztésében nagy része volt Jeremy Hylton-nak.

A Mailman honlapjai

<http://www.gnu.org/software/mailman>

<http://www.list.org>

Tüköroldalak:

<http://www.list.org>

<http://www.gnu.org/software/mailman>

<http://mailman.sf.net>

A Mailman 2.1 működéséhez Python 2.1.3 vagy újabb szükséges, amelyet a következő címről lehet letölteni:

<http://www.python.org>

A megfelelő működéshez ajánlott a Python 2.3.3 verziója, amely a dokumentáció készítésekor (2003. december 31.) a legújabb verzió volt. A program a 2.2-es Python sorozat legfrissebb verziójával, a 2.2.3-as verzióval is működik. Azonban nem működik Python 2.0 vagy régebbi verziókkal együtt.

Szükség van továbbá egy ANSI C fordítóra; a gcc (a GNU C fordítója) megfelel a célnak. A Mailman jelenleg GNU/Linux és más Unix-típusú operációs rendszereken (pl. Solaris, *BSD, MacOSX, stb.) képes futni. Nem működik Windows rendszeren, azonban bármely platform web és levelező programja képes a Mailman-nel együttműködni.

Az INSTALL.hu állományban részletesen lehet olvasni a telepítéséről. Ha régebbi verziójú Mailman-t frissítünk, akkor a fontosabb tudnivalók érdekében olvassuk el az UPGRADING.hu állományt is.

TULAJDONSÁGOK

A NEWS állományban a 0.9-es verziótól fogva követhetők nyomon a változások. A TODO állományban a kívánság lista található (azon újabb tulajdonságok, változtatások, amelyeket a felhasználók szívesen használnának). A következő címen működés közben látható a Mailman 2.1:

<http://www.python.org/mailman-21/listinfo>

A Mailman rendelkezik mindazokkal a szükséges tulajdonságokkal, amelyekkel egy levelezőlista-kezelő programnak manapság rendelkeznie kell és még annál is többel:

- Web alapú lista adminisztrálás majdnem minden beállításhoz. Web alapú feliratkozás és felhasználói beállítási lehetőség. Minden listához külön megváltoztatható információs oldal.

- Biztonsági beállítások, mint moderálás, nyitott és zártkörű listák, feliratkozási korlátozások, privát tagnévsor és a beküldések feladók szerinti korlátozása.
- Beépített automatikus web alapú levél archiválás mind nyilvános, mind privát archívumhoz, valamint külső archiváló használatának lehetősége.
- Felhasználóként változtatható levélküldési mód, MIME formátumú vagy RFC 1153 típusú sima szöveges digestek kézbesítése.
- Beépített levél/hírcsoport átjáró.
- Beépített automatikus válaszadás.
- Majordomo-stílusú e-mail parancsok.
- Beépített és bővíthető visszapattnás kezelés.
- Beépített spamkezelés és MIME-típusú levelek szűrése.
- Bővíthető levélkézbesítési mechanizmus.
- Virtuális tartományok támogatása.

KÖVETELMÉNYEK

Bármilyen olyan levél továbbító programmal együtt lehet használni a Mailman-t, amelyik támogatja a közvetlen SMTP kapcsolatot a 25-os porton keresztül. Azonban néhány MTA-nál (pl. Exim és Postfix) a Mailman képes levelezőlisták web felületről történő létrehozására vagy törlésére.

A Mailman bármilyen CGI/1.1-et támogató web kiszolgálón képes működni. A létrehozott HTML oldalai egyszerűek, kevés grafikai

elemet tartalmaznak így szinte bármilyen web-böngészőn és internet kapcsolaton futnia kell. Az oldalak megtekinthetőségét, kinézetét időnként IE 5.5, Windowsos Netscape 4.7x és Mozilla, valamint Linuxos Netscape 4.7x és Mozilla böngészőkkel (esetenként Linuxos Lynx-el és MacOS-os Netscape-el is) teszteljük.

Root-jogokkal kell rendelkezünk azon a gépen, ahova a Mailman-t telepíteni szeretnénk. A szükséges beállításokról az INSTALL.hu állományban lehet olvasni.

A Mailman web és e-mail felülete minden web-böngészővel és levelező-programmal képes együtt működni, de ajánlott hogy a levelezőprogram tudja kezelni a MIME-típusú leveleket. Nincs szükség Java, JavaScript vagy egyéb plugin használatára.

ELSŐ SAJÁT LEVELEZŐLISTA LÉTREHOZÁSA

A következőkben tárgyalunkhoz szükséges egy az INSTALL.hu alapján telepített és beállított Mailman. Az első saját levelezőlistánk létrehozásához a következőt kell tenni:

- Először váltsunk abba a könyvtárba, ahova a Mailman telepítve lett (alap esetben ez az /usr/local/mailman), majd állítsuk be a rendszer adminisztrátori jelszót az alábbi paranccsal

(Megjegyzés: a kijelzett üzenetek eltérhetnek, ha a Mailman alapértelmezett nyelve eltér a magyartól. A magyar nyelvi támogatás beállításához parancssori programok esetén a LANG környezeti változót kell 'hu'-ra állítani (pl. bash esetén ez az export LANG=hu paranccsal tehető meg). A webfelület a \$MAILMAN_HOME/Mailman/mm_cfg.py állományba elhelyezett DEFAULT_SERVER_LANGUAGE=hu sorral mindenhol magyarul fog megjelenni.)

```
% bin/mmsitepass
új webhely jelszó: [sajátjelszó]
Ismételd meg a jelszót: [sajátjelszó]
Jelszó megváltoztatva.
```

- Látogassunk el a következő címre:

<http://sajatgep.vhol/mailman/create>

Töltsük ki az oldalon található űrlapot az ott található magyarázatok segítségével. A "Listalétrehozó jelszó" mezőbe a fent beállított jelszót adjuk meg. "A lista első gazdájának e-mail címe" mezőbe írjuk be a saját e-mail címünket, majd válasszuk ki a lista tulajdonosának értesítését.

- Kattintsunk a "Lista létrehozása" gombra.

- A postafiókunkban hamarosan találunk egy értesítést a Mailmantól az új levelezőlista sikeres létrehozásáról.
- MEGJEGYZÉS: Exim vagy Postfix-től eltérő MTA-k használata esetén (pl. Sendmail vagy qmail) szükség van még a levelezőlisták bejegyzéseinek manuális hozzáadására a rendszer bizonyos állományaihoz. Kövessük a kapott e-mailben leírtakat (fontos tudnunk milyen MTA található a rendszerünkön, kérjük olvassa el az Ön MTA-jához tartozó README állományt).
- Ezek után látogassunk el a lista adminisztrációs oldalára (ide vagy a főoldalról elindulva vagy a Mailmantól kapott levélben megadott cím alapján juthatunk el). Az internet cím valahogy így fog kinézni:

`http://sajatgep.vhol/mailman/listinfo/listank`

Adjuk meg a listához használt jelszavunkat és kattintsunk a "Belépés..." gombra.

- Kattintsunk a "Listatagok kezelése" majd a "Több tag felvétele" hivatkozásra.
- Adjuk meg a saját e-mail címünket a nagy üres szövegmezőben, majd kattintsunk a "Változtatások mentése" gombra.
- Ezek után a listank@sajatgep.vhol címre küldjünk egy próba levelet. A Mailman a levelünket percekben belül továbbítani fogja címünkre.

Gratulálunk! Sikeresen létrehoztuk és teszteltük első saját levelezőlistánkat. Probléma esetén olvassuk el az alább található EGYÉB INFORMÁCIÓK részt.

Az online dokumentáció elérhető a kicsomagolt Mailman könyvtárban a

`file:admin/www/index.html`

címen.

Chris Kolar lista-tulajdonosoknak készült használati útmutatója a következő címen található:

`http://www.imsa.edu/~ckolar/mailman/`

Ezen felül számos levelezőlista segíthet bármilyen, a Mailman-nal kapcsolatos probléma megoldásában.

Mailman-Users

Mailman felhasználók nyitott listája, ahova a telepítéssel, használattal és egyéb problémákkal, vagy kérdésekkel lehet fordulni. A mély technikai témák nem ennek a listának a területe.

`http://mail.python.org/mailman/listinfo/mailman-users`

Listowners

A levelezőlista segítséget nyújthat azoknak a lista tulajdonosoknak és szerkesztőknek, akik nem rendelkeznek "shell eléréssel" a Mailman programot futtató levelezőlista-szerverhez. Nem területe a listának a mély technikai témák megvitatása.

`http://listowner.org`

Mailman-Announce

Egy csak-olvasható lista az újabb verziók megjelenésének és más fontos hírek bejelentésére.

`http://mail.python.org/mailman/listinfo/mailman-announce`

Mailman-Developers

Nyitott lista mindazoknak, akik Mailman jövőbeni fejlődésében, fejlesztésében szeretnének részt venni. A listán mély technikai témák kerülnek megvitatásra.

<http://mail.python.org/mailman/listinfo/mailman-developers>

Mailman-I18N

Nyitott lista a Mailman nyelvi támogatásának koordinálására. A 2.1-es Mailman teljesen több-nyelvű lesz.

<http://mail.python.org/mailman/listinfo/mailman-i18n>

Mailman-Checkins

Csak-olvasható lista, ahol a nyilvános anonymous CVS tár változásai olvashatóak. A listára feliratkozva mindig nyomon lehet követni a Mailman fejlesztését.

<http://mail.python.org/mailman/listinfo/mailman-checkins>

A Mailman projektet a SourceForge-on a következő címen lehet elérni:

<http://sf.net/projects/mailman>

A SourceForge-on keresztül célszerű a különböző hibákat bejelenteni (bugs report) és hibajavításokat közzétenni.

KÖSZÖNETNYILVÁNÍTÁS

Szeretném megköszönni a segítségét Funk Gábornak, Fintha Attilának és Komáromi Tamásnak, akik remek tanácsaikkal és észrevételeikkel segítették a fordítás elkészültét.

Magyar fordítás: Vizi Szilárd és csapata.

INSTALL

Ebben az állományban a GNU Mailman telepítéséhez találunk útmutatót, amely a GNU szabványos autoconf programjával történik. Legelőször a rendszeren kell bizonyos beállításokat (ezekről alább lehet olvasni) elvégezni, s csak ezek után kell a Mailman programot konfigurálni és telepíteni.

FONTOS: A crontab beállításra fordítsunk különös figyelmet, ahogy az a lentebb le van írva, különben a Mailman nem fog megfelelően működni.

FRISSÍTÉS: A frissítés legtöbbször nem jelent mást, mint a régi verzióra egy új verzió másolását. Azonban frissítés előtt ajánljuk elolvasásra az UPGRADING állományban található fontos információkat.

A GNU Mailman honlapjának címe: <http://www.list.org>

A programot le lehet tölteni a következő címről:

http://sf.net/project/showfiles.php?group_id=103

0. Telepítési igények

Rendelkeznünk kell egy levelezőszerverrel (MTA), ahová a leveleket lehet küldeni, és egy webszerverrel, amely támogatja a CGI/1.1 API-t. Webszervernek az Apache megfelelő választás, míg MTA-nak a Postfix, az Exim, a Sendmail és qmail közül bármelyik megfelel.

Ezenfelül egy ANSI C fordítóval is rendelkezünk kell, amelynek segítségével lehet a biztonsági követelményeknek megfelelő wrapper programokat előállítani. GNU C fordítója, a gcc 2.8.1 vagy ennél frissebb megfelel a célnak. A gcc beszerzéséről bővebben a következő oldalon olvashatunk:

<http://gcc.gnu.org>

A Python feldolgozó nyelvnek is telepítve kell lennie a rendszeren. A megfelelő működéshez jelenleg a Python 2.1.3-as vagy 2.2.1-es verziója ajánlott. Bővebb információt a Python forráskódjának, vagy RPM csomagjának és lefordított bináris változatának beszerzéséről a következő oldalon olvashatunk:

<http://www.python.org>

Ha a Python-t forráskódjából szeretnénk telepíteni, akkor az Unix-típusú operációs rendszereknél elegendő a hagyományos parancsokat, `./configure; make install` kiadni. A `"make test"` futtatásánál jó pár tesztelési lépést át lesz ugorva -- de ne aggódjunk, ezekre a funkciókra valószínűleg nem is lesz szükségünk. A Mailman csak azokat az alapvető funkciókat használja, amelyek minden gépen megtalálhatóak.

Ha találunk a levelezőszerverünkhöz (MTA) README.<MTA-nk> állományt, akkor olvassuk el azt. A Mailman egyes MTA-kkal képes fejlettebb együttműködésre (pl. listák automatikus létrehozása, megszüntetése webfelületről). Ilyen levelezőszerver az Exim vagy a Postfix. A levelezőszerverhez szükséges beállításokról a megfelelő README állományokban lehet olvasni.

1. A rendszer beállítása

A következő lépésekhez root jogokkal kell rendelkezniük.

A Mailman program telepítése előtt a rendszerhez kell adni a megfelelő felhasználót és csoportot.

- Hozzunk létre egy új felhasználót `'mailman'` névvel. Ez legtöbbször az `/etc/passwd` állományba kerül bejegyzésre. Ha a `'mailman'` felhasználónév már foglalt, akkor bármilyen más egyedi nevet

választhatunk, ekkor olvassuk el a `--with-username` kapcsolónál található megjegyzést.

- Hozzunk létre egy `'mailman'` nevű csoportot. Ez legtöbbször az `/etc/group` állományba kerül bejegyzésre. A Mailman állományai a `'mailman'` csoport tulajdonában kerülnek telepítésnél, `set-group-id` bit jogokkal. Ezzel mind a webes, mind a levelezőprogramok együtt tudnak működni. Ha a `'mailman'` csoportnév már foglalt, akkor bármilyen más egyedi nevet megadhatunk. Ekkor olvassuk el alább a `--with-groupname` kapcsolónál található megjegyzést. Az előző lépésben létrehozott mailman felhasználó tagja kell, hogy legyen ennek a csoportnak.
- Hozzunk létre egy könyvtárat, ahová a programot telepíteni fogjuk (ezt a továbbiakban `$prefix`-nek nevezzük). A Mailman összes állománya a `$prefix`-ben megadott könyvtárba fog kerülni. A `"configure --help"` paranccsal a különböző telepítési kapcsolókról lehet többet megtudni.

A Mailman 2.1 alapértelmezett telepítési könyvtára az `/usr/local/mailman`. Mailman 2.1alpha2-nél régebbi verzióknál ez a `/home/mailman` könyvtár volt. Az alapértelmezett beállítást a `configure` parancs `--prefix` kapcsolójával lehet megváltoztatni. Ha Mailman 2.1-nél régebbi verziót szeretnénk frissíteni, akkor külön meg kell adnunk a `--prefix` kapcsolót, hacsak nem akarjuk a listákat magunktól áthelyezni (ez előnyös lehet, ha saját magunk szeretnénk a frissítést elvégezni).

Néhány rendszeren az `/usr/local` könyvtárat tartalmazó partíción tiltva van a SUID programok futtatása. A Mailman rendes működéséhez `'set-gid'`-es programokat használ, ezért az ilyet nem támogató partíciókra nem célszerű telepíteni a programot.

Ha a rendszer ilyen partícióval rendelkezik, akkor a Mailman-t olyan könyvtárba kell telepíteni, ahol engedélyezett a `setgid` programok futása.

Győződjünk meg arról hogy a telepítés könyvtára a `mailman' (vagy a --with-groupname-nél megadott) csoport tulajdonában van és rendelkezik a setgid bit-tel (BSD rendszergazdák olvassák el a README.BSD állományt bővebb információért). A könyvtárat ezenfelül mindenki számára elérhetővé és futtathatóvá kell tenni. Ezt pl. a következő parancsokkal lehet biztosítani:

```
% cd $prefix
% chgrp mailman .
% chmod a+rx,g+ws .
```

Ezek után lehet a Mailman-t telepítésre felkészíteni, majd telepíteni.

2. configure futtatása

A --with-mail-gid ÉS --with-cgi-gid KAPCSOLÓKRA VALÓSZÍNŰLEG SZÜKSÉGÜNK LESZ EZÉRT OLVASSUK EL A HOZZÁJUK TARTOZÓ LEÍRÁST!

A következőkben leírt parancsokhoz nem kell root jogokkal rendelkezni. A saját bejelentkezési nevünk alatt, vagy azon névvel, amellyel általában telepítjük a programokat is végrehajthatjuk a következő parancsokat. Nem szükséges mailman felhasználóként sem a lépéseket követni, de akár mailman-ként is dolgozhatunk. Azonban győződjünk meg róla, hogy az alkalmazott felhasználóinév tagja-e a mailman csoportnak, mivel az előző lépésben létrehozott könyvtárhoz írási joggal rendelkezik a mailman csoport.

Fontos, hogy a telepítés könyvtárához legyen írási jogunk, valamint setgid beállítási lehetőség (NFS és más köteteket be lehet állítani úgy, hogy ne vegyék figyelembe a setgid beállításokat).

Ha már telepítettünk más GNU programot, akkor a configure szkript

ismerős lehet. Általában a program forráskönyvtárában kell lépni és a configure parancsot kiadni kapcsolók nélkül:

```
% cd mailman-<verzió>
% ./configure
% make install
```

A következő kapcsolókkal lehet a Mailman telepítést befolyásolni.

`--prefix=<dir>`

Hagyományos GNU configure kapcsoló, amellyel megadható a könyvtár neve, ahova a Mailmant telepítjük. Alap esetben a \$prefix értéke /usr/local/mailman. A könyvtárnak léteznie kell és a fenn leírtaknak megfelelően beállítva kell lennie.

`--exec-prefix=<dir>`

Hagyományos GNU configure kapcsoló, amellyel a rendszerre jellemző lefordított (bináris) állományok telepítési helyét lehet megadni.

`--with-var-prefix=<dir>`

Változó állományok, mint a list archívuma és a beállításait tartalmazó konfigurációs állományok a <dir> könyvtárba kerüljenek, ne pedig a prefix-ben vagy exec_prefix-ben megadottba.

`--with-python=</path/to/python>`

Egyéni Python parancsfeldolgozó elérési útjának megadása a wrapper programok számára. Alap esetben a burok \$PATH környezeti változójában legelőször megtalált parancsfeldolgozó ez. Jó tudni, hogy a szkriptek parancssorbóli futtatásakor mindig a \$PATH alapján legelőször megtalált Python parancsértelmező kerül felhasználásra.

--with-username=<felhasználónév-vagy-uid>

Megadja, hogy milyen a `mailman'-tól eltérő felhasználónév legyen az alapértelmezett. Csak akkor használjuk, ha a `mailman' felhasználónevet már más valaki használja (pl. Mark Ailman bejelentkező neve). Név helyett annak azonosító számát (user id) is meg lehet adni. Fontos, hogy a \$prefix-ben megadott könyvtár az itt megnevezett felhasználó tulajdonában legyen.

--with-groupname=<csoporthív-vagy-gid>

Megadja, hogy milyen a `mailman'-tól eltérő csoportnév legyen az alapértelmezett. Csak akkor használjuk, ha a `mailman' csoportnév már foglalt. Név helyett annak azonosító számát (group id) is meg lehet adni. Fontos, hogy a \$prefix-ben megadott könyvtár az itt megnevezett csoport tulajdonában legyen.

--with-mail-gid=<csoporthív-vagy-csoporthívok>

Egyéni csoportot lehet megadni a szkriptek futtatásához a mail wrappernél. <csoporthív-vagy-csoporthívok>-nak több csoportazonosító számot vagy csoport nevet is meg lehet adni. A felsoroltak közül a legelső, amelyik létező csoportot ad meg kerül felhasználásra. Alap esetben a lista értéke `mailman other mail daemon'.

Ez beállítás rendszerenként változik, így minden rendszeren ellenőrizni kell, mivel a csoport azonosítója biztonsági okok miatt rögzítve lesz a wrapper programban. Sendmail-t használó rendszereken a sendmail.cf állomány "DefaultUser" sora tartalmazza a kívánt azonosítót. (Ha nincs megadva, csak megjegyzésként feltüntetve /commented out/, akkor is valószínű a jó értéket mutatja...)

Az itt megadandó értékhez olvassuk el a saját MTA-nk dokumentációját vagy konfigurációs állományát.

`--with-cgi-gid=<csoport-vagy-csoportok>`

Egyéni csoportot lehet megadni a szkriptek futtatásához a CGI wrappernél. `<csoport-vagy-csoportok>`-nak több csoportazonosító számot vagy csoport nevet is meg lehet adni. A felsoroltak közül a legelső, amelyik létező csoportot ad meg kerül felhasználásra. Alapesetben a lista értéke ``www www-data nobody'`.

A megfelelő érték web kiszolgálókként eltérő lehet. Ezért minden rendszeren ellenőrizni kell az értékét, mert a csoport azonosítója biztonsági okok miatt rögzítve lesz a CGI wrapper programban. Rossz érték esetén egyetlen Mailman CGI szkript sem fog futni.

Apache használók a `httpd.conf` állomány ``Group'` részénél találják meg a megfelelő értéket.

`--with-cgi-ext=<kiterjesztés>`

A `cgi-bin` programok kiterjesztését lehet vele megadni. A CGI programok a `$PREFIX/cgi-bin` könyvtárba kerülnek a megadott kiterjesztéssel ellátva (néhány web kiszolgálónál kötelező kiterjesztést is megadni). A `<kiterjesztés>`-ben a pontot is meg kell adni.

`--with-mailhost`

A megadott teljes gépnevet fogja részben felhasználni a program a kimenő e-mailekhez. Telepítés után a beállított értéket a `$PREFIX/Mailman/mm_cfg.py` állományon keresztül módosítani lehet.

`--with-urlhost`

A megadott teljes gépnevet fogja részben felhasználni a program az internetes címekhez. Telepítés után a beállított értéket a `$PREFIX/Mailman/mm_cfg.py`

állományon keresztül módosítani lehet.

--with-gcc=no

A fordításhoz ne gcc-t használjon, még ha az meg is található a rendszeren. Ilyenkor a \$PATH-ből elérhetőnek kell lennie a `cc` fordítóprogramnak.

3. Telepítés utáni ellenőrzés

A telepítés után ellenőrizni kell a jogokat és tulajdonosokat, ehhez használjuk a check_perms programot:

- cd \$prefix

- bin/check_perms

Ne próbáljuk meg a forrás könyvtárból a bin/check_perms programot futtatni; az csak a telepített könyvtárban (lásd \$prefix) fog megfelelően működni.

Ha a program hiba nélkül fut le, akkor nagy valószínűséggel jól van beállítva a telepített Mailman :) Ha hibaüzenetet ír ki, akkor a hibát lehet lépésről lépésre, újratelepítéssel, vagy a check_perms program segítségével (ez a legjobb megoldás) kijavítani:

- Be kell lépni azon felhasználóként, akivel a telepítés történt (és aki tulajdonosa a \$prefix-ben található állományoknak), vagy rootként; és a következőket kell végrehajtani:

- bin/check_perms -f parancs kiadása

- Majd ismételd meg az előző lépéseket, amíg a hiba meg nem szűnik!

4. Végző beállítások

Gratulálunk! Sikeresen telepítettük a Mailman programot. A működéséhez már csak a web részt és levelezőrészt kell a Mailman-nel egybehangolni.

- Ha a levelezőrendszert és web kiszolgálót eltérő gépeken szeretnénk futtatni, a Mailman programot NFS-en megosztani, akkor fontos, hogy a gépek rendszerórája egymással szinkronban legyen. A Mailman/LockFile.py állományban a CLOCK_SLOP konstans állítgatásával a különböző rendszerek közötti időeltérést lehet ellensúlyozni.
- Állítsuk be a web kiszolgálót, hogy a \$prefix/cgi-bin könyvtárban található CGI-eket futtassa. Ezt valószínű root-ként kell megtenni.

A következőkhöz hasonló sort kell majd megadni (a \$prefix helyére a tényleges könyvtár nevet kell megadni teljes elérési úttal):

```
Exec          /mailman/*      $prefix/cgi-bin/*  
vagy:  
ScriptAlias   /mailman/       $prefix/cgi-bin/
```

Részletekért olvassuk el a web kiszolgálónk dokumentációját.

- Fontos meggyőződnünk arról, hogy az a felhasználói azonosító, amellyel a CGI fut *nem* tagja-e a `mailman' csoportnak, különben a privát archívumot nem lehet majd elérni.
- A Mailman, Python és GNU logókat másoljuk a web kiszolgáló által is elérhető könyvtárba. Pl. Apache-nál van egy `icons' könyvtár, ahova érdemes a megosztott képeket másolni. A képeket a következő paranccsal tudjuk a megfelelő könyvtárba másolni:

```
% cp $prefix/icons/*.{jpg,png} /apache/eleresi/utvonala/icons
```

Ezek után a \$prefix/Mailman/mm_cfg.py állományban lehet megadni

a logók elérését. Valahogy így:

```
IMAGE_LOGOS = '/images/'
```

Az IMAGE_LOGOS alapértelmezett értéke '/icons/'. Bővebben a Defaults.py-ben lehet róla olvasni.

- A levelezőlisták nyilvános Pipermail archívumához a web kiszolgálónak a következőt kell megadni:

Apache-nál például:

```
Alias /pipermail/ $varprefix/archives/public/
```

ahol \$varprefix többnyire a \$prefix, ha a configure-nak nem lett külön megadva a --with-var-prefix kapcsoló.

Részletekért olvassuk el a web kiszolgálónk dokumentációját. A kiszolgálót be kell állítani a szimbolikus kötések követésére is különben a Pipermail archívumot nem lehet elérni. Apache felhasználóknak a FollowSymLinks opciót kell megadniuk.

Ha a levelezőlistáink nyilvános archívumait különböző nemzeti nyelveken is helyesen szeretnénk megjeleníteni, akkor fontos, hogy kapcsoljuk ki a webszerverünk alapértelmezett karakterkódolását a Pipermail könyvtárra különben az archívum nem fog megfelelően megjelenni. Ha az alapértelmezett telepítési útvonalat használtuk, akkor Apache esetén a következőhöz hasonló sort kell elhelyezni a konfigurációs állományában:

```
<Directory "/usr/local/mailman/archives/public/">  
    AddDefaultCharset Off  
</Directory>
```

Ezek után indítsuk újra a web kiszolgálót.

- Hozzuk létre a "központi" levelezőlistát. A jelszó emlékeztetők ennek a listának a címéről fognak érkezni. A lista neve általában "mailman", de igény szerint változtatható a MAILMAN_SITE_LIST változó segítségével, amelyet az mm_cfg.py (lásd később) állományban kell megadni.

```
% bin/newlist mailman
```

Kövessük az utasításokat és bővebb információkért olvassuk el a README.hu állományt.

Ezek után állítsuk be a központi levelezőlistát. Ebben segítségünkre lehet a központi listák általános beállításait tartalmazó sablonfile, a misc/sitelist.cfg állomány. A sablont a központi levelezőlistára a következő paranccsal lehet alkalmazni:

```
% bin/config_list -i data/sitelist.cfg mailman
```

Mielőtt felhasználnánk a sablont mindenképp nézzük meg, hogy az abban található beállítások megfelelnek-e nekünk (annak ellenére, hogy számos beállítást érintetlenül hagy a sitelist.cfg fájl). Ha módosítottuk a központi listát, akkor utána el ne felejtsük az adminisztrációs oldalon is leellenőrizni a beállított értékeket, hogy megfelelnek-e az elvárásainknak.

Ezek után iratkozzunk fel erre a mailman levelezőlistára az adminisztrációs oldalon keresztül. Az e-mailben történő feliratkozás ezen a ponton még nem működik, így azt ne használjuk.

- A crontab beállítása. A Mailman működéséhez számos folyamatot kell a cron-nak futtatnia. Régebbi verziójú Mailman frissítésekor ne felejtsük el az új crontab beállításokat elvégezni. Ha egyidejűleg több Mailman-t szeretnénk a rendszerünkön futtatni, akkor ügyeljünk a crontab helyes beállítására, mivel a hibás

crontab beállítások a párhuzamosan működő Mailman-eknél hibához vezethet.

Ha a crontab támogatja az `-u` kapcsolót, akkor a következő lépéseket root-ként hajtsuk végre. A `$prefix/cron/crontab.in` felvételéhez a crontab-ba a következő parancsokat kell végrehajtani:

```
% cd $prefix/cron
% crontab -u mailman crontab.in
```

Ha használtuk a `--with-username` kapcsolót, akkor a mailman helyett az ott megadott értéket használjuk az `-u` kapcsolónál. Ha crontab nem támogatja az `-u` kapcsolót, akkor használjuk a következő parancsokat:

```
% cd $prefix/cron
% su - mailman
% crontab crontab.in
```

- A Mailman grunner démon futtatásához a következő parancsot kell kiadni a `$prefix` könyvtárban:

```
% bin/mailmanctl start
```

Ha azt szeretnénk, hogy minden rendszerindításkor elinduljon a program, és az operációs rendszerünk támogatja a `chkconfig` parancsot (pl. RedHat és Mandrake Linux), akkor rootként elég a következő parancsokat kiadni:

```
% cp scripts/mailman /etc/init.d/mailman
% chkconfig --add mailman
```

(Néhány rendszernél az `/etc/init.d` helyett `/etc/rc.d/init.d` -t kell használni.)

Debian-t használók a chkconfig helyett a

```
% update-rc.d mailman defaults
```

parancsot használják.

Azon Unix rendszereknél, amelyek nem támogatják a chkconfig programot másoljuk át a scripts/mailman állományt az init könyvtárba:

```
% cp misc/mailman /etc/init.d
```

és hozzuk létre a szimbolikus kötéseket rootként a következő parancsokkal:

```
% cd /etc/rc.d/rc0.d
% ln -s ../init.d/mailman K12mailman
% cd ../rc1.d
% ln -s ../init.d/mailman K12mailman
% cd ../rc2.d
% ln -s ../init.d/mailman S98mailman
% cd ../rc3.d
% ln -s ../init.d/mailman S98mailman
% cd ../rc4.d
% ln -s ../init.d/mailman S98mailman
% cd ../rc5.d
% ln -s ../init.d/mailman S98mailman
% cd ../rc6.d
% ln -s ../init.d/mailman K12mailman
```

- Ellenőrizzük, hogy a Defaults.py állományban a DEFAULT_EMAIL_HOST és DEFAULT_URL_HOST változók helyes értéket mutatnak. Ha nem, akkor a helyes értékkel ezeket a változókat az mm_cfg.py állományba helyezzük el. Megjegyezzük, hogyha ezen két változó valamelyikét módosítjuk, akkor lehet hogy szükséges az mm_cfg.py állományba ezen

változók után még megadni a következő parancsot is:

```
add_virtualhost(DEFAULT_URL_HOST, DEFAULT_EMAIL_HOST)
```

5. Mailman finomhangolása

A következő beállításokat a 2. fejezetben tárgyalt felhasználóként hajtsuk végre.

- A `$prefix/Mailman/Defaults.py` számos beállítást tartalmaz. Ha ezek valamelyikét meg akarjuk változtatni, akkor azt a `$prefix/Mailman/mm_cfg.py` állományban kell megadni és SOHA SEM a `Defaults.py` állományban. A `Defaults.py`-ben minden beállításról bővebb információ található. Egyes beállítások megváltoztatásának nincs hatása már létrehozott listáknál, ha szeretnénk a már létező listákra is alkalmazni a módosított beállításokat, akkor újra létre kell hozni az adott listát. A listákat az adminisztrációs oldalon keresztül, vagy parancssorban a `bin/withlist` vagy `bin/config_list` parancs segítségével lehet beállítani.

A telepítés nem írja felül a létező `mm_cfg.py` állományt, így abban nyugodtan el lehet helyezni a szükséges személyes beállításokat.

Fontos: Soha `*ne*` változtassuk meg a `HOME_DIR` vagy `MAILMAN_DIR` változók értékét. Ezeket automatikusan a telepítő állítja be.

Magyar nyelv ékezetes betűinek helyes megjelenítéséhez az alábbi sorokat helyezzük el az `mm_cfg.py` állományban:

```
DEFAULT_CHARSET='iso-8859-2'  
VERBATIM_ENCODING=['iso-8859-2']
```

- Ha azt szeretnénk, hogy a rendszer alapértelmezett nyelve a magyar legyen, akkor adjuk az `mm_cfg.py`-hez a következő sort:

```
DEFAULT_SERVER_LANGUAGE='hu'
```

- Rendszerszintű jelszó létrehozása a következő paranccsal történik:

```
% $prefix/bin/mmsitepass <rendszerszintű-jelszó>
```

Ezt a jelszót meg lehet adni, mind felhasználó-, mind admin-szintű beállítások megváltoztatásánál. A rendszer adminisztrátor így képes a listák beállításának módosítására is.

Itt lehet megadni a rendszerhez használatos "lista létrehozó" jelszót is (ezzel lehet majd a rendszerszintű jelszó mellett listákat létrehozni és törölni web-felületről). A jelszó beállításához a "mmsitepass"-t a -c kapcsolóval kell futtatni.

6. Munka megkezdése

Lásd a README.hu állomány "ELSŐ SAJÁT LEVELEZŐLISTA LÉTREHOZÁSA" szakaszát egy próba levelezőlista létrehozásához.

7. Problémák

Ha hibát észlelünk a Mailman működése során, akkor először olvassuk el a "Gyakori hibák" részt. Ha ott nem találunk megoldást, akkor nézzük meg a FAQ.hu állományt és az Interneten az online FAQ Wizard-ot.

Ezek után ellenőrizzük a \$prefix/logs/error állományban a Mailman bejegyzéseket.

A naplóállományok helye rendszerenként változik. Lehet az a /var/log/maillog, vagy a /var/log/syslog. Számos gépen a napló-állományok az /adm/log/ könyvtárban és nem a /var/log könyvtárban

találhatóak.

Hiba esetén küldjük el a hiba leírását a `mailman-users@python.org` címre. Mondjuk el, hogy mit csináltunk a hiba előtt és hogy mit tartalmaznak ezzel kapcsolatban a naplóállományok. Ezen túl adjuk meg a gépünk paramétereit és a használt Python, Mailman verziószámát.

8. Gyakori hibák

Hiba: Az összes Mailman oldalra a "404 File not found" üzenetet kapom.

Megoldás: Nincs megfelelően beállítva a web kiszolgálója a Mailman CGI parancsok futtatására. Győződjünk meg a következőkről:

- 1) A web kiszolgáló be van állítva a `$prefix/cgi-bin` futtatására
- 2) A web kiszolgáló megfelelően lett újraindítva.

Olvassuk el a web kiszolgálónk dokumentációját ezen módosítások végrehajtásáról.

Hiba: Az összes Mailman oldal "Internal Server Error" jelez.

Megoldás: A legvalószínűbb, hogy a CGI szkripteknek rossz GID vagy UID lett megadva. Nézzük meg a naplóállományokat. Ha a következő sorhoz hasonlót találunk benne:

```
Attempt to exec script with invalid gid 51, expected 99
```

akkor újra kell telepíteni a Mailmant és a `$CGI-GID`-nek 51-et adjunk meg a telepítés részben ismertettek szerint.

Hiba: A listára küldött levelem visszajön azzal az üzenettel, hogy

a lista nem található!

Megoldás: Valószínűleg nem lettek a szükséges bejegyzések a rendszer alias adatbázisához adva a newlist parancs futtatása után. Ha mégis hozzá lettek adva, akkor valószínű nem lett frissítve az adatbázis, vagy nem lett futtatva a newaliases parancs. Ekkor az 5. fejezetben tárgyaltak szerint járjunk el.

Hiba: A listára küldött levelem visszajön a következő üzenettel:
"unknown mailer error".

Megoldás: A legvalószínűbb, hogy nem megfelelő GID vagy UID lett megadva a levelezőrésznek. Nézzük meg a naplóállományokat. Ha a következőhöz hasonlót találunk benne:

```
Attempt to exec script with invalid gid 51, expected 99
```

akkor újra kell telepíteni a Mailmant és a \$MAIL_GID-nek 51-et adjunk meg, ahogy az a telepítési részben le van írva. Postfix felhasználók olvassák el a következő hibát is, amely náluk fordul elő leggyakrabban.

Hiba: MTA-nak Postfixet használók és a mail wrapper programok hibás GID-et említenek a naplóállományokban.

Megoldás: A Postfix main.cf állományában az "alias_maps" változóknál hozzunk létre egy új alias állomány bejegyzést. Ezt az új állományt a Mailman könyvtárába, vagy olyan helyre ahol a Mailman-nak írási joga is van helyezzük el;
mailman felhasználóként futtassuk a Postfix "postalias" parancsát ezen az állományon.

```
% postalias <alias_fájl_neve>
```

Még mindig mailman felhasználóként adjuk ki a következőt:

```
% python -c'import os; print os.getgid()'
```

Ezzel megkapjuk azt a csoport azonosító számot, amelyet meg kell adnunk a Mailman telepítésekor a mail wrapper programok számára. Legyen ez mondjuk "joGID". Konfiguráljuk újra a Mailman-t a következő paranccsal:

```
% ./configure --with-mail-gid=joGID
```

Olvassuk el a README.POSTFIX.hu állományt a Postfix és Mailman összekapcsolásához.

Hiba: A listára küldött levelem visszajön a következő üzenettel:
"sh: mailman not available for sendmail programs"

Megoldás: A rendszer sendmail korlátozott shell-t használ (smrsh). Be kell állítani az smrsh-t úgy, hogy egy szimbolikus hivatkozást hozunk létre a mail wrapper-ről (%prefix/mail/mailman) abba a könyvtárba, amelyben az engedélyezett smrsh-k futhatnak.

Ezt a könyvtárat általában a /var/admin/sm.bin,
/usr/admin/sm.bin vagy /etc/smrsh helyen találhatjuk.

Debian Linux alatt a megadott /usr/lib/sm.bin könyvtár nem működik, ezért hozzuk létre az /usr/admin/sm.bin könyvtárat és ide hozzuk létre a szimbolikus hivatkozásokat.

Hiba: Összezavarodtam a configure futtatásánál. Hogyan tudom törölni a felesleges fájlokat újratelepítéshez?

Mailman - The GNU Mailing List Management System
Copyright (C) 1998-2004 by the Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA

```
Megoldás: % make clean
           % ./configure --with-jó-kapcsolók
           % make install
```

ÁLTALÁNOS BEÁLLÍTÁSOK

A Mailmannek minden gond nélkül működnie kell egy általánosan beállított Postfix-el. A dokumentáció írásakor a beállítások Postfix 19991231 verziótól kezdve pl13 verzióig, és 200010228 verziótól pl08 verzióig, valamint Postfix 2.0.15-tel lettek tesztelve.

A Postfix alapértelmezésként speciális módon kezeli az -owner és -request címekre érkező leveleket. Ha azt szeretnénk, hogy az ide érkező leveleket is a Mailmannek továbbítsa a program kapcsoljuk ki ezt a funkciót a main.cf állomány módosításával:

```
owner_request_specil = no
```

Ahhoz hogy esetlegesen használni tudjuk a Mailman VERP küldési képességét ki kell kapcsolnunk az luser_relay (ez az alapértelmezett) beállítást és módosítanunk kell a recipient_delimiter értékén. Ehhez kommentezzük ki a main.cf állományban található összes luser_relay sort, visszatérve ezzel az alapértelmezett opcióhoz. Továbbá, helyezzük el main.cf fájlba a következő sort:

```
recipient_delimiter = +
```

A + jel használatával a a Defaults.py állományban megadott alapértelmezett VERP_FORMAT és VERP_REGEX értékek megfelelően fognak működni.

A Postfix 450-es hibaüzenettel válaszol, ha egy nem létező lokális címre próbálunk meg levelet továbbítani. Mivel ez nem jelent végzetes hibát, ezért a Mailman a kézbesítést addig ismételgeti, amíg le nem jár a DELIVERY_RETRY_PERIOD változóban megadott idő (alap esetben 5 nap). Ennek elkerülésére be lehet állítani a Postfixet úgy, hogy a nem létező helyi felhasználóknak történő

kézbesítéskor a nem végzetes hibaüzenet helyett egy végzetes hiba üzenetet küldjön. Ehhez a beállításához a következő sort kell elhelyezni a main.cf állományba:

```
unknown_local_recipient_reject_code = 550
```

Végül ha Postfix-stílusú virtuális tartományokat használunk, akkor olvassuk el az erről szóló lejjebb található fejezetet.

POSTFIX ÉS MAILMAN ÖSSZEKAPCSOLÁSA

A Postfixet össze lehet kapcsolni a Mailmannel úgy, hogy egy új lista létrehozásakor, vagy törlésekor a Postfix alias adatbázisa automatikusan frissüljön. Ehhez a következők lépéseken kell végigmennünk.

A leírás feltételezi, hogy a Mailmant az alapértelmezett telepítési helyére telepítettük, ez a /usr/local/mailman. Ha nem így tettük, akkor a lépéseknél megfelelően módosítsuk az elérési útvonalakat azokra amiket a konfigurálásakor a --prefix és --with-var-prefix kapcsolóknál használtunk.

- Ha Postfix-stílusú virtuális tartományokat használunk és szeretnénk, hogy a Mailman kezelje ezeket a tartományokat, akkor először olvassuk el a virtuális tartományokról lejjebb található fejezetet.

- A \$prefix/Mailman/mm_cfg.py állományban adjuk meg a következőt:

```
MTA = 'Postfix'
```

Az MTA változó értékével arra a programmodulra hivatkozunk a Mailman/MTA könyvtárban, amelyik az adott MTA-ra jellemző hívások segítségével kezeli létrehozáskor vagy törléskor a listákat.

- Keressük meg a Defaults.py állományban a POSTFIX_ALIAS_CMD és POSTFIX_MAP_CMD változókat. Győződjünk meg, hogy a változók a postalias és postmap programok helyes teljes elérési útvonalát tartalmazzák. Fontos, hogy ha ezeket megszeretnénk változtatni, akkor ne itt tegyük hanem az mm_cfg.py állományban.
- Futtassuk a genaliases programot az aliases állomány létrehozásához:

```
% cd /usr/local/mailman
% bin/genaliases
```

Győződjünk meg, hogy a data/aliases és data/aliases.db tulajdonosa a `mailman` és az állományok csoport tulajdonosa is a `mailman`.

```
% su
% chown mailman:mailman data/aliases*
```

- Módosítsuk a Postfix main.cf konfigurációs állományában az alias_maps változót úgy, hogy a tartalmazza a következő elérési útvonalat is:

```
/usr/local/mailman/data/aliases
```

(nincs a végén .db kiterjesztés) Az alias_database változóban ne adjuk meg ezt az útvonalat, mert nem szeretnénk hogy módosítsa a Postfix newaliases parancsa a Mailman aliases.db állományát. Mindössze azt szeretnénk, hogy a Postfix figyelembe vegye az aliases.db állományt, amikor a hely kézbesítési címeket keresi.

Célszerű egy hash: típusú adatbázisként megadni ezt az állományt. Ezt megtehetjük például így:

```
alias_maps = hash:/etc/postfix/aliases,
            hash:/usr/local/mailman/data/aliases
```

- A Mailman konfigurálásakor használjuk a `--with-mail-gid=mailman` kapcsolót (mellesleg ez az alapértelmezett beállítás), mivel az `aliases.db` állomány tulajdonosa a `'mailman'` felhasználó és a Postfix a Mailman wrapper programját `mailman uid` és `gid` alatt próbálja meg futtatni.

Ennyi! Egy kis gond: amikor létrehozunk, vagy törölünk egy listát, akkor az `aliases.db` állomány azonnal frissül, viszont nem fut le automatikusan a változtatások érvényesítéséhez a `"postfix reload"` parancs. Ez azért van így, mert ehhez root jogokkal kell rendelkezni, vagy egy `suid-root` programnak kell futnia, amelyek biztonsági kockázatot jelentenek. A megoldás egyetlen hátrányát tehát az jelenti, hogy körülbelül csak egy perc múlva veszi észre a Postfix a módosításokat és érvényesít a változtatásokat. Ez a kis kényelmetlenség szerintünk elviselhető.

VIRTUÁLIS TARTOMÁNYOK

A Postfix 2.0 támogatja "virtual alias domains" kezelést, amelyet a korábbi Postfix verziókban egyszerűen csak Postfix-típusú virtuális tartománykezelésnek hívtak. Ahhoz, hogy a Mailman képes legyen a virtuális tartománykezelés használatához mind a Mailman, mind a Postfix beállításokon változtatni kell. A Mailman alapesetben az összes virtuális tartományhoz tartozó bejegyzést az `/usr/local/mailman/data/virtual-mailman` állományban tárolja. A `postmap` program is ezen állomány alapján készíti el a `virtual-mailman.db` állományt, melyet a Postfix valójában használni fog.

Először a Postfix dokumentációban foglaltak alapján állítsuk be a virtuális tartománykezelést (bővebben a Postfix `virtual(5)` súgója ad eligazítást). Fontos, hogy el ne felejtsük megadni, ahogy a súgóban le van írva a `"virtual-alias.domain anything"` sort, mivel a Mailman `virtual-mailman` állománya nem tartalmazza ezt a sort. Javasoljuk, hogy még mielőtt telepítenénk a Mailmant, győződjünk meg róla, hogy megfelelően működik a virtuális tartománykezelés.

A következő lépésként a Postfix `virtual_maps` változóhoz kell egy újabb könyvtár elérési utat adni:

```
virtual_maps = <your normal virtual files>,  
    hash:/usr/local/mailman/data/virtual-mailman
```

Esetünkben a Mailman az alapértelmezett telepítési könyvtárban helyezkedik el. Régebbi Postfix-eknél nincsen `virtual_alias_maps` változó, ez esetben használjuk a `virtual_maps` változót.

Következő lépésként az `mm_cfg.py` állományba állítsuk be a `POSTFIX_STYLE_VIRTUAL_DOMAINS` változóba azon virtuális tartományokat, amelyekre a Mailmannek figyelnie kell. Előfordulhat, hogy nem itt nem soroljuk fel a Postfix beállításunk által támogatott összes virtuális tartományunkat. Az itt megadott értékeknek pontosan egyezniük kell a levelezőlisták `host_name` változójában megadott értékekkel.

Nézzük meg egy példán keresztül:

Tegyük fel, hogy a Postfixben a `dom1.ain`, `dom2.ain` és `dom3.ain` virtuális tartományokat szeretnénk kezelni. Ehhez mondjuk a `main.cf` állományban a következő beállítások vannak megadva:

```
myhostname = mail.dom1.ain  
mydomain = dom1.ain  
mydestination = $myhostname, localhost.$mydomain  
virtual_maps =  
    hash:/some/path/to/virtual-dom1,  
    hash:/some/path/to/virtual-dom2,  
    hash:/some/path/to/virtual-dom2
```

Példánkban a `virtual-dom1` állományban a következő sorokat adtuk meg:

```
dom1.ain  IGNORE
```

`@dom1.ain @mail.dom1.ain`

Ezzel a Postfixet arra utasítottuk, hogy minden `dom1.ain`-re címzett levelet tegyen ugyanabba a postafiókba, mint a `mail.dom1.ain`, az alapértelmezett címre érkezőket.

Ez esetben a `POSTFIX_STYLE_VIRTUAL_DOMAINS`-ben nem szabad megadnunk a `dom1.ain`-t, mivel különben a Mailman levelezőlisták bejegyzései a `dom1.ain` tartományban a következő lennének

```
mylist@dom1.ain      mylist
mylist-request@dom1.ain mylist-request
# és így tovább ...
```

Az egyre pontosabb bejegyzések felülírják az előttük található általánosabb bejegyzéseket, ezzel megakadályozzák a `dom1.ain` levelezőlisták megfelelő működését.

Azonban az `mm_cfg.py` állományban megadjuk a `dom2.ain` és `dom3.ain` értékeket:

```
POSTFIX_STYLE_VIRTUAL_DOMAINS = ['dom2.ain', 'dom3.ain']
```

Ezzel elértük, hogy minden olyan listának, amelyet a Mailman a fenti két tartomány valamelyikében hoz létre a megfelelő bejegyzései a `/usr/local/mailman/data/virtual-mailman` állományban helyesen fognak megjelenni.

Csak úgy mint a fent említett `data/aliases*` állományoknál, itt is győződjünk meg arról, hogy a `data/virtual-mailman` és `data/virtual-mailman.db` állományok felhasználó- és csoporttulajdonosa `'mailman'` legyen. A megfelelő működéshez tehát előbb hozzuk létre a virtuális tartományokat, futtassuk a `bin/genaliases` programot és végül ellenőrizzük ezen állományokon a jogosultságokat. Ezen túl, többet már nem kell velük foglalkoznunk.

EGY MÁSIK MEGKÖZELÍTÉS

Fil <fil@rezo.net> máshogy, virtual maps és szabványos keresési kifejezések használatával kezeli a virtuális tartományokat.

Leírása megtalálható:

(franciául) <http://listes.rezo.net/comment.php>

(angolul) <http://listes.rezo.net/how.php>

Egy jó (és egyszerűbb) módszer azon esetekben ha nem bánjuk, hogy a tartomány részben egy újabb hostnév jelenik, ezzel növelve annak a címnek a karakter hosszát, amelyen keresztül a felhasználók elérik a listát. Pl. a listát a felhasználók sajátlista@lists.dom.ain címmel érik el sajátlista@dom.ain cím helyett.

Ezt a megoldást a dokumentáció készítője alaposan nem tesztelte.

UPGRADE

MEGJEGYZÉS A MODERÁLÁSHOZ

2.0.x verzióról 2.1 verzióra történő frissítéskor ellenőrizzük, hogy a moderálási és a privát beállítások nem térnek-e el a korábban beállított értékektől. A moderálási és privát beállítások, a könnyebb érthetőség és kezelhetőség érdekében, jelentősen megváltoztak a Mailman újabb verziójában. Hiába azonban az igyekezet, hogy minél tökéletesebben, problémamentesen kerüljenek át a régi, összetett beállítások az új rendszerbe, mégis előfordulhat, hogy a beállítások átvétele hibás lesz.

Különösen a (Privát beállítások -> Feladók szűrése) `default_member_moderation`, `generic_nonmember_action`, és `accept_these_nonmembers` beállításokat ellenőrizzük le. Ezenfelül célszerű ellenőriznünk a Listatagok kezelése menüben a felhasználók egyenkénti moderálási állapotát is.

FRISSÍTÉS KORÁBBI VERZIÓKRÓL

A Mailman frissítése többnyire nem jelent mást, mint egy újabb verzió telepítését a létező telepített verzióra. Azonban néhány esetben saját magunknak kell bizonyos változtatásokat elvégeznünk.

Azt hogy egész pontosan mit kell csinálnunk az függ attól, hogy melyik verzióról melyik verzióra állunk át. Mindegyik esetben először kapcsoljuk ki az e-mail és web hozzáférést a telepített Mailmanhez, mivel lényegében egy adatbázis frissítünk és nem lenne szerencsés ha frissítés közepén az adatbázisunk megváltozik.

A következőket javasoljuk :

- Kapcsoljuk le a bejövő levelekért felelős mail deamont. A legtöbb smtp kiszolgáló megpróbálja később továbbítani nekünk a leveleket, ha lezártuk a 25-ös portot.
- Átmenetileg kapcsoljuk ki a web hozzáférést is a telepített Mailmanhez. Ezt elérhetjük úgy, hogy vagy ideiglenesen leállítjuk a web kiszolgálót, vagy létrehozunk egy "átmenetileg szünetel" oldalt a Mailman URL-ökhöz. Bővebb információkért olvassuk el a web kiszolgálónk dokumentációját.

Működő listák sablonállományait nem frissíti a Mailman. Hogy ilyen esetben mit kell csinálni, azt Chuq Von Rospach leírásából lehet megtudni a következő címen:

<http://mail.python.org/pipermail/mailman-users/2000-September/006826.html>

[Valójában MM2.1a2 verzióra történő váltáskor a program lecseréli a sablonállományokat, azokat pedig törli amelyek megegyeznek az eredeti változattal (az összehasonlítást az md5 ellenőrzőösszegek alapján végzi).]

FRISSÍTÉS 2.0.x VERZIÓRÓL 2.1 VERZIÓRA

A Mailman 2.1-es verziójában drasztikus változtatáson esett át a grunner rendszer. A grunnert többé nem cron-ból kell indítani! Helyette a bin/mailmanctl program indításával vagy leállításával lehet kezelni a levelek feldolgozását. A program egyben egy Unix indító szkript is. Fontos, hogy el ne felejtsük frissíteni a crontab bejegyzést az új cron/crontab.in állománnyal.

MEGJEGYZÉS: Nagyon fontos, hogy ***MIELŐTT*** frissítenénk MM2.1alpha2 előtti verzióról MM2.1alpha2-nél újabb verzióra, akkor hagyjuk hogy a régi grunner folyamat a qfiles/ könyvtárban található összes kézbesítésre váró üzenetet feldolgozza, mert a frissítés

után már nem fogja feldolgozni ezeket az üzeneteket az új grunner.

MEGJEGYZÉS: Mailman 2.1beta1-nél újabb verzióra való átálláskor újra létre kell hoznunk az aliases állományokat, mivel az újabb verziókban a wrapper program neve megváltozott mailman-re. A README.<MTank>.hu állományokban részletes leírás található a Mailman és az adott levelezőszerver összekapcsolásáról.

Az aliases állományt a bin/genaliases programmal könnyen újra létre lehet hozni.

A 2.1-es Mailman már többféle nyelven is használható, támogatja az eltérő karakterkészleteket. Régebbi verziókban listánként mindössze egy nyelv volt használható és az is az angol volt. A frissítés során minden egyes lista lists/<listanév> könyvtárába létrehoz egy `en' nevű könyvtárat a program. A frissítés során a lists/<listanév> könyvtárakban található .txt és .html állományokat bemásolja a program a lists/<listanév>/en könyvtárba.

Ha módosítottuk a sablonokat, hogy ne (csak) angol szöveget tartalmazzanak, akkor saját magunknak kell átnevezni az `en' könyvtárat a használt nyelv kódjának megfelelő nevű könyvtárrá. A Mailman frissítéseket végző programja automatikusan törli azokat a sablonokat, amelyek több, azonos példányban is megtalálhatóak, de nem árt személyesen is átfutnunk a sablonállományok listáját ellenőrzésképpen.

Ha 2.0.x-es rendszert használunk nem a szokványos javításokkal, akkor a frissítés során problémákba ütközhetünk. Ilyenek lehetnek:

- Ha a #413752 (mindig sima szövegformátum) javítást telepítettük, akkor a frissítés nem fog gond nélkül zajlani. A #651406 frissítés segíthet a probléma megoldásában.

http://sf.net/tracker/?group_id=103&atid=300103&func=detail&aid=413752

http://sf.net/tracker/?group_id=103&atid=300103&func=detail&aid=651406

LISTÁK EGYENKÉNTI FRISSÍTÉSE

Ha félünk a 2.1-es verzióra történő teljes átállásból eredő problémáktól, akkor megtehetjük hogy a listáinkat egyenként frissítjük az újabb verzióra. Ehhez mindössze egy üres könyvtárba kell telepítenünk a Mailman 2.1-es verzióját, erre a könyvtárra \$MM21 -ként fogunk a későbbiekben hivatkozni. (A 2.0-ás verzió könyvtárára pedig a továbbiakban \$MM20 -ként hivatkozunk.)

Ilyen esetben a Mailman 2.0 és 2.1-es verziója egyszerre fog működni a rendszerünkön addig, amíg teljes egészében át nem állunk a 2.1-es verzióra. Az általunk használt MTA és web kiszolgálóktól függően ez a módszer gond nélkül, simán is működhet, azonban előfordulhatnak komoly problémák is.

Ha az Apache kiszolgálónál a mod_rewrite funkciót tudjuk használni, akkor beállíthatjuk, hogy mind a 2.0-ás és 2.1-es Mailman ugyanazt a /mailman és /pipermail címet használhassa; ezzel elérhetjük hogy a lista adminisztrátorok, a felhasználók zavartalanul tudják használni a rendszert.

Minden egyes listánál, amelyet a másik verzióba akarunk átvinni a következőket tegyük.

* Állítsuk le az MTA-t.

Ha a kimenő forgalmunk számottevő, akkor megtehetjük, hogy úgy állítjuk be az MTA-t, hogy csak a 127.0.0.1 (localhost) címről érkező kapcsolatokat fogadja, így a 2.0-ás Mailman a várakozó leveleket kézbesíteni tudja. Hogy ezt a beállítást, hogyan tudjuk megtenni az függ a használt MTA-tól; Exim esetén a "local_interfaces = 127.0.0.1" sort kell megadnunk, majd

"kill -HUP" paranccsal újraindítanunk az Exim démont.

* Állítsuk le a webkiszolgálót. Jobb megoldás, ha csak a /mailman/ oldalakhoz érkező kéréseket irányítjuk át egy "átmenetileg nem elérhető" oldalra, ettől még más oldalakat el fognak tudni érni a felhasználók.

A megoldás itt is programfüggő; Apache esetén a mod_rewrite segítségével az alábbi módon oldható meg:

```
RewriteRule ^/mailman/.*/var/www/unavailable.html [L]
```

(Természetesen előbb létre kell hoznunk a /var/www/unavailable.html oldalt.)

* Kényszerítsük a 2.0-ás Mailmant, hogy dolgozza fel a várakozó leveleket a következő paranccsal:

```
python -S $MM20/cron/qrunner
```

(Ezt csak akkor kell megtennünk, ha a \$MM20/qfiles könyvtár nem üres, azonban győződjünk meg ekkor, hogy az MTA képes fogadni kapcsolatot a 127.0.0.1 címről.)

* Mozgassuk át a listát:

```
cd $MM20
mv -i lists/foo-list $MM21/lists
mv -i archives/private/foo-list $MM21/archives/private
mv -i archives/private/foo-list.mbox $MM21/archives/private
rm archives/public/foo-list
rm archives/public/foo-list.mbox
cd $MM21
bin/withlist -l -r fix_url mylist
```


(Az utolsó lépés, a `fix_url` használata csak akkor szükséges, ha a 2.0-ás és 2.1-es verziók eltérő URL-t használnak.)

* Módosítsuk a web kiszolgáló beállítását, hogy a listák oldalai elérhetőek legyenek. Két megoldás lehet; az egyszerűbb az, hogy egy új címen keresztül érjük el a 2.1-es verziót, pl. `/mailman-21`. Ehhez az Apache `mod_rewrite` modulját kell használnunk:

```
RewriteRule /mailman/(.*)/(foo-list.*) /mailman-21/$1/$2 [R=temp]
```

(A `[R=temp]` rész azt jelenti, hogy a `/mailman-21/` cím csak átmeneti és ha már minden listát átmozgattunk a 2.1-es verzióba, akkor megszűnik és az összes listát a `/mailman/` címen lehet majd elérni.)

A másik megoldásnál nem szeretnénk egy új címet használni, hanem mind a 2.0-ás, mind a 2.1-es verzió listáit ugyanazon a címen keresztül szeretnénk elérni. A megoldás ekkor az Apache `mod_rewrite` moduljával a következő lehet:

```
RewriteRule ^/mailman/(.*)/(foo-list.*) \  
$MM21/cgi-bin/$1/$2 \  
[T=application/x-httpd-cgi]
```

Ezen megoldás másik előnye, hogy gyorsabb is, mivel nem történik átirányítás.

Bármelyik megoldást is alkalmazzuk el ne felejtkezzünk a lista archívumának az átirányításáról sem:

```
RewriteRule ^/pipermail/(foo-list.*) $MM21/archives/public/$1
```

* Indítsuk újra a web kiszolgálót (vagy kapcsoljuk ki az átirányítást, amely az "átmenetileg szünetel" oldalt hozza be).

* Indítsuk újra az MTA-t (vagy állítsuk be, hogy mostantól már ne csak a 127.0.0.1 címről fogadjon kapcsolatot).

FRISSÍTÉS 2.0 VERZIÓRÓL 2.0.x VERZIÓRA (AHOL $x \geq 1$)

Nem kell sok mindent tenni, a "make install" -lal a frissítés is megtörténik.

FRISSÍTÉS 2.0 béta VERZIÓRÓL 2.0 végleges VERZIÓRA

ÚJRA le kell futtatnunk a configure programot; a config.status újrafuttatása sajnos az autoconf programban történt változások miatt nem elegendő. A config.status első sorai között megtalálhatjuk, hogy régebben milyen beállításokkal futtattuk le a configure-t.

A végleges 2.0-ás verzióban a cron feladatok és azok gyakorisága megváltozott. A `mailman` felhasználónak újra be kell tölteni a misc/crontab.in fájlból a helyes beállításokat. Bővebben erről az INSTALL dokumentációban lehet olvasni.

HA KIHAGYJUK EZT A LÉPÉST, AKKOR A MAILMAN NEM FOG MEGFELELŐ HATÉKONYSÁGGAL MŰKÖDNI.

FRISSÍTÉS 1.x VERZIÓRÓL 2.x VERZIÓRA

Erősen javasolt, hogy győződjünk meg a frissítés előtt, hogy a Mailman feldolgozási sora üres.

A 1.x verzióban a levelek kézbesítését a run_queue program végezte. A 2.x verziókban ez a program megszűnt (funkcióját

az MTA vette át), és jelenleg nem ismert, hogy milyen hatást idéz elő a frissítés ezen a programrészen, de valószínű hibás működéshez vezetne.

Ha a `$prefix/data` könyvtár üres, akkor a Mailman feldolgozási sora biztosan üres. Ha a könyvtár "mm_q." kezdetű fájlokat tartalmaz, akkor még mindig van kézbesítésre váró levél a feldolgozási sorban. A `$prefix/cron/run_queue` program indításával kényszeríteni lehet ezen levelek kézbesítését. A program többszöri indítása nem sietteti a feldolgozás idejét, mivel a program párhuzamos feldolgozások előtt zárolja a leveleket. Fontos megjegyeznünk, hogy a feldolgozási sor kiürítése időbe kerül és a rendszert erősen terhelheti (ezért is lett átírva a kézbesítés a 2.x verzióban).

Nem kell használni a "make update" parancsot, ha 1.0 vagy 1.1-ről 2.0-ára frissítünk, mert ezt a parancsot a "make install" automatikusan lefuttatja. Viszont frissítenünk kell a crontab bejegyzéseket, hogy ezentúl ne a `cron/run_queue`, hanem a `cron/qrunner` program legyen időszakosan elindítva. Ezek után nyugodtan lehet törölni a `$prefix/cron/run_queue` fájlt.

Ha egy 1.0 béta előtti verzióról szeretnénk frissíteni, akkor azt a lejjebb található módon végezzük.

FRISSÍTÉS PRE-1.0 VERZIÓRÓL 2.x VERZIÓRA

Az 1.0 béta előtti verziókról történő frissítéskor legelőször a Mailman könyvtár rendszerét kell frissíteni, ezt két módon tehetjük meg.

Első módszernél a forrás könyvtárában miután kiadtuk a "make install" parancsot, adjuk ki a "make update" parancsot. Ekkor létrejön egy "update.log" nevű állomány a forrás gyökér-

könyvtárába. Ha a program a Mailman fájlrendszer frissítésekor olyan problémába ütközik, amelyet nem tud megoldani, akkor ebbe az "update.log" állományba fogja menteni a hibaüzenetet. Célszerű ezért ezt a fájlt frissítés után átnéznünk.

A frissítést végrehajthatjuk úgy is, hogy belépünk a telepített Mailman könyvtárába (pl. \$prefix) és futtatjuk a bun/update programot. Ez a program ugyanazt hajtja végre, mint az előbbi, de nem hozza létre az update.log fájlt.

Ellenőrizzük a crontab beállításokat. Töröljük a szükségtelen, elavult programok indítására vonatkozó bejegyzéseket, elsősorban a cron/upvolumes_yearly, cron/upvolumes_monthly, vagy cron/archive programokra utaló bejegyzéseket.

A "MAKE UPDATE" MŰKÖDÉSE

A továbbiakban a "make update" működéséről, magyarázatokkal el látva olvashatunk. Reméljük, hogy ez segít az esetleges problémák elhárításában.

Jó tudni, hogy nem jelenthet problémát, ha minden egyes frissítéskor kiadjuk a "make update" parancsot, azonban az 1.0-nál újabb verziók esetén nem fog változást hozni!

- 1.0b10 verzióra történő frissítéskor a templates/options.html fájlt át kell másolni minden egyes listánál a lists/<listanév>/könyvtárba. Ha módosítottuk az options.html fájlt - mondjuk a webfelületen keresztül -, akkor a változtatásokat saját magunknak kell végrehajtani az új fájlokon.
- 1.0b7 verzióra történő frissítéskor a Mailman/smtplib.py{,c} állományokat törölni kell, a funkcióját a Python 1.5.2 verzióban található smtplib veszi át.

- Az archívum helye az 1.0b6-os telepítésével megváltozik, mivel ebben a verzióba a Piplermail már be lett építve. A teendők,

1) ha a listának csak privát mbox archívuma van, akkor a \$prefix/archives/private/<listanév> átkerül a \$prefix/archives/private/<listanév>.mbox/<listanév> helyre,

2) ha a listának csak nyilvános mbox archívuma van, akkor a \$prefix/archives/public/<listanév> átkerül a \$prefix/archives/private/<listanév>.mbox/<listanév> helyre

és egy szimbolikus hivatkozást kell létrehozni, a \$prefix/archives/public/<listanév>.mbox hivatkozásnak a \$prefix/archives/private/<listanév>.mbox/<listanév> helyre kell mutatnia.

3) ha a listának mindkét típusú archívuma létezik már, akkor a "make update" a kettő közül attól függően azt választja, hogy a lista éppen nyilvános vagy privát archívummal rendelkezik. Ezek után a mások mbox-ot átnevezi mbox.preb6 -á.

4) ha a lista olyan CVS verziót használ, ahol az archívum helye a \$prefix/public_html/archives volt, akkor a program ezeket a \$prefix/archives/private/<listanév> helyre mozgatja át és létrehozza a \$prefix/archives/public/<listanév> szimbolikus hivatkozást, ha a lista archívuma nyilvános. Ezzel egy jogosultsági probléma is megoldódik.

A régi listák archívumának létrehozásához lépünk be `mailman' felhasználóként és futassuk a következő parancsot:

```
$prefix/bin/arch <listanév> <mbox-archívum-elérési-útvonala>.
```

Továbbá a beta6 alapértelmezés szerint az archívumot mind mbox, mind html formátumban létrehozza. Hogy csak egyik, vagy

mindkettő vagy semelyik módszer szerint se archiváljon az a megfelelő helyen beállítható. Erről bővebben a `$prefix/Mailman/Defaults.py` állományban lehet olvasni.

A fejlesztések során volt egy olyan rövid időszak, amikor az archiválást végző kód nem csak a saját csomagján belül volt elhelyezve. Ekkor az archívumba elhelyezendő levelekhez a HyperArch modulra is szükség volt, amelynek azóta a helye megváltozott. A problémát a következő paranccsal lehet megoldani:

```
ln -s $prefix/Mailman/Archiver/HyperArch.py \  
    $prefix/Mailman/HyperArch.py
```

- Ha 1.0b4 -nél régebbi verzióról frissítünk, akkor a "make update" a lista-specifikus sablonokat (`$prefix/templates/<listanév>/*`) minden egyes listánál áthelyezi a `$prefix/lists/<listanév>` könyvtárba. Ellenőrizzük, hogy a `$prefix/templates` könyvtárban maradó általános sablon fájlok közül bármelyik is meg változott-e. (Elméletileg csak az `options.html` változik meg a b5-ről b6 verzióra történő átálláskor.)

Nagyon régi Mailman verzióknál még `<listanév>` alkönyvtár sem található a `$prefix/templates` könyvtárban! Ez esetben saját magunknak kell bizonyos fájlokat átmásolni az új könyvtárba. A következő parancs átmásolja a szükséges fájlokat:

```
cp      templates/{archives,handle_opts,listinfo,roster,subscribe}.html  
lists/<listanév>
```

- Törölni kell azokat a modulokat, amelyek a korábbi verziókban megtalálhatóak voltak, de az újabbakban le lettek cserélve, vagy új nevet kaptak.

Megjegyzés: A Mailman GYIK (angolul a FAQ) már on-line is elérhető a "FAQ Wizard" rendszer segítségével, a <http://www.python.org/cgi-bin/faqw-mm.py> címen.

GYAKRAN ISMÉTLŐDŐ KÉRDÉSEK

K. Hogyan kell helyesen leírni a program nevét?

V. A "Mailman" első "M" betűjét nagybetűvel kell írni, míg a második "m" kicsivel írandó. A "MailMan" írásforma helytelen (mg. felejtsük el itt a szótagok kezdőbetűjének nagybetűs írásformáját).

K. A kimenő leveleket iszonyúan lassan továbbítja a program. Úgy tűnik, hogy amikor az MTA valamely címzettnél DNS kérést végez, akkor a grunner nagyon lassan dolgozza fel a várakozási sort. Ötlet?

V. Valóban az MTA DNS kérést végez a címzetteknel, amikor a leveleket helyben továbbadja (pl. a Mailman az SMTPDirect.py-vel az MTA-nak). Ez bizony hiba. Ki kell kapcsolnunk a gép folyamatos DNS kérését.

Exim esetén, ez a receiver_verify_hosts megfelelő beállításával történhet. Bővebb információt a README.EXIM.hu-ban találhatunk. Más MTA-k esetében (természetesen) más beállítást kell végrehajtani. Először olvassuk el az MTA-hoz tartozó megfelelő README állományt, majd végső esetben az MTA dokumentációját.

K. A lista tagok a Mailman List-* fejléceiről kérdezgetnek. Mik ezek valójában?

V. Ezeket a fejlécek a Mailman minden egyes kimenő levélhez hozzáadja, a felhasználók kényelmében. A Mailman az RFC 2369-ben foglaltaknak megfelelően hozza létre ezeket a fejléceket. Az RFC 2369 ezeket a fejléceket definiálja. Ha még mindig kíváncsiak a felhasználók, akkor olvassuk el jó tanácsokért a README.USERAGENT.hu állományt.

Végső esetben a lista adminisztrációs oldalán az Általános beállítások részben le lehet tiltani ezen fejlécek hozzáadását a kimenő levelekhez.

K. Hogyan tudom a felhasználó címét megjeleníteni a minden levél alján megtalálható láblécben?

V. A Mailman 2.1-es verziótól támogatja a személyre szabott levélküldést, azonban alapesetben nincs engedélyezve, mivel túlzott terhelést okozhat a rendszeren. Ha szeretnénk támogatni az egyedi levelek küldését, akkor a `~mailman/Mailman/mm_cfg.py` állományban adjuk meg a következő beállításokat:

```
VERP_PASSWORD_REMINDERS = 1
VERP_PERSONALIZED_DELIVERIES = 1
VERP_DELIVERY_INTERVAL = 1
VERP_CONFIRMATIONS = 1
```

A VERP (variable envelope return path = változtatható válaszcímű levél) engedélyezése után a lista adminisztrációs oldalán a Nem-Digest küldési beállítások részben ismertetett változók segítségével a felhasználó feliratkozási címét, nevét, beállítási oldalát és jelszavát is el lehet helyezni.

K. A felhasználóim kerülni szeretnék a HTML formátumú leveleket és én biztonsági okok miatt szeretném a MIME csatolt állományokat is a levelekből eltávolítani. Hogyan tehetem ezt meg?

V. A Mailman 2.1-ben lehetőség van a tartalomszűrésre, amellyel korlátozni, vagy tiltani lehet hogy milyen típusú mellékletek, levelek jelenhetnek meg a listán. Ehhez szükséges beállításokat a lista adminisztrációs oldalán keresztül a Tartalom szűrés részben lehet módosítani.

A 2.1 előtti Mailman verziókhoz külső programokat, minnt pl. a demime vagy stripmime kell használni. Ezekről bővebb információt a következő oldalakon lehet találni:

(Stripmime) <http://www.phred.org/~alex/stripmime.html>

(Demime) <http://scifi.squawk.com/demime.html>

- K. Mi van akkor, ha "document contains no data" (üres dokumentum) üzenetet kapok a webkiszolgálótól, vagy a levelek nem kerülnek feldolgozásra, vagy "Premature end of script" (hibás szkript befejezés) vagy "Mailman CGI error!!!" (Mailman CGI Hiba) üzenetet kapok.
- V. A leggyakoribb hibát az okozza, hogy a C wrapper program nem a webkiszolgáló által várt GID jogokkal lett lefordítva. Ugyanilyen hibát okoz az is, ha a levelezőrendszer más GID-del hívná meg a lefordított C levél wrappert.

A hiba elhárításához újra kell fordítani a Mailman-t a `--with-cgi-gid` és `--with-mail-gid` kapcsolókkal. Az INSTALL állományban erről bővebben lehet olvasni.

Ezek a hibaüzenetek nem a Mailman naplóállományokban, hanem a rendszerszintű naplóállományokban jelennek meg. A CGI wrappernél felmerülő hibák közvetlenül a web böngészőben jelennek meg, a szükséges GID-del együtt. Ez nagymértékben tud segíteni a probléma elhárításánál.

A syslog-ot be lehet állítani, hogy pl. a mail.error hibákat egy megadott állományba naplózza; például Solaris rendszeren a

```
mail.debug                /var/log/syslog
```

sor azt jelenti, hogy az üzenetek a /var/log/syslog állományba kerülnek. (A rendszerhez tartozó syslog.conf állomány mondja meg

(ha van), hogy az üzenetek hova kerüljenek. A syslog man-jában erről bővebb információ található.)

Ha a rendszerünk így van beállítva és a mailman/listinfo lap megtekintésénél UID vagy GID hibával találkozunk, akkor a /var/log/syslog állományban találhatjuk meg a kívánt és a működéshez szükséges UID/GID értékeket.

K. Miért 'fagynak' le a weboldalak?

V. Az ok, hogy a CERN típusú webserverek a Python folyamatokat futási állapotban hagyhatják, ezzel lefagyaszthatják a CGI-eket. A probléma kiküszöbölhető Apache használatával.

Előfordulhat, hogy lejárt zárolások vannak jelen. A Mailman az adatbázisának biztonsága érdekében szigorúan veszi a zárolást, de néha rendszerhibák következtében lejárt zárolások maradhatnak. A \$prefix/locks könyvtárban találhatóak a zároló állományok. (Könnyen megállapítható, hogy lejárt-e valamelyik zárolás, ha a ps paranccsal megnézzük, hogy az állományban található folyamatazonosítóhoz (PID) tartozik-e futó folyamat. Amennyiben nem tartozik, akkor az az állomány nyugodtan törölhető.)

K. Mit kell időnként megnéznem?

V. A szkriptek többsége a ~mailman/logs/error állományba jegyzi be hibaüzeneteit, így alkalmanként ebben kell keresnünk hibára utaló üzenetet.

Az állományban *nem* találhatóak meg a szintaktikai hibára vonatkozó üzenetek, mivel ezek az installálás során, a .py állományok fordításánál azonnal megjelennek. Szintaktikai hibák előfordulhatnak a forrás nem megfelelő módosításakor, vagy 'nem modul' típusú szkriptek használatakor.

A ``compile'` vagy ``compileall'` Python modulokkal bármikor le lehet bájtonként fordítani egy állományt, vagy egyszerűen a Python értelmező segítségével azt betölteni és tesztelni.

K. Miért nem működik az archívum?

V. A listára érkezett már levél? Ez egy ismert hiba; az archívum addig nem működik amíg legalább egy levél nem érkezett a listára.

K. Rendben, az archívum működne, de mégsem tudom a nyilvános archívumot elérni. Miért?

V. Apache esetén győződjünk meg, hogy a FollowSymlinks a nyilvános archívum útvonalára is meg van-e adva. Fontos tudni, hogy az archívum mindig a privát könyvtárban található; nyilvános archívumnál mindössze egy hivatkozás mutat a privát archívumra. Bővebb információ olvasható a következő címen:

<http://mail.python.org/pipermail/mailman-users/1998-November/000150.html>

K. Még mindig nem megy? QMail-t használ?

V. Győződjünk meg róla, hogy a "mailman" wrapper program meghívásánál a "preline" meg van adva:

```
|preline /home/mailman/mail/mailman post listname
```

A "preline" használatával egy Unix-típusú "From " fejléc jön létre a levelekben, amelyek az archiváláshoz szükségesek. Az archívum mbox állományában lévő minden üzenetbe a következő sort beszúrva

```
From somebody Mon Oct 9 12:27:34 MDT 2000
```

megoldódik a probléma. Futtassuk újra a "bin/arch listaneve" parancsot. Az archívumnak most már létre kell jönnie. További információt a

README.QMAIL állományban lehet olvasni.

K. Még mindig nem megy? GNU/Linux-ot használ?

V. Olvassuk el a README.LINUX.hu állományt.

K. Az archívumból szeretnék pár levelet törölni. Hogyan tehetem ezt meg?

V. David Rocher megoldása:

- * töröljük a \$prefix/archives/private/<listaneve> könyvtárat
- * szerkesztővel módosítsuk a következő állományt:
\$prefix/archives/private/<listaneve>.mbox/<listaneve>.mbox
- * futtassuk a \$prefix/bin/arch <listaneve> parancsot

K. Igenre állítottam a "member_posting_only" (csak_tagok_küldhetnek) beállítást, hogy csak a listatagok küldhessenek levelet a listára, azonban úgy néz ki, hogy a listatagoktól érkező összes levél engedélyezésre vár a megjelenéshez. Miért?

V. Egyes rendszereken a levél feladója (pl. a Unix "From " sor) hibás lehet. Ekkor a Mailman a feladót nem tudja listatagként azonosítani. 1.0b12-es verzióig a Mailman alapesetben előbb a levél feladóját és nem a From: mezőben található feladót keresi meg, mivel az előbbit az SMTP program tölti ki, míg utóbbinak a felhasználó bármit megadhat.

[A levél feladójának megváltoztatásából adódó hibák gyakran előfordulnak, de a sendmail "owner-alias" szolgáltatásról illik itt pár szót ejteni:

Ha egy levél érkezik a "foo" listára, és az "owner-foo" alias is meg van adva, akkor a levél feladója egyszerűen "owner-foo"-ként lesz azonosítva.

A Mailman 1.0rc2 verziójától fogva már megfelelően kezeli ezt a (nem változtatható) sendmail-es problémát. Régebbi verziók esetében megoldást jelenthet az, ha az ajánlott "owner-LISTNAME" sort kihagyjuk minden egyes Mailman listánál az alias állományból.

Azonban ilyen probléma esetén biztosabb megoldást jelent, ha a From: fejléct használjuk a levél feladójaként beállított helyett. Ehhez az mm_cfg.py állományba kell a következő sort elhelyezni:

```
USE_ENVELOPE_SENDER=0
```

Ha még (vitathatóbb) biztonságra akarunk törekedni akkor az mm_cfg.py állományba a következőt írjuk be:

```
USE_ENVELOPE_SENDER=1
```

Azonban olvassuk el a Defaults.py-ben található leírást a változóról. Alapesetben a 2.0-s Mailman a From: fejléct használja fel cíamazonosításhoz.

K. Mennyire biztonságos a Mailman web azonosítási módszere?

V. Ha a Mailmant SSL-re képes web kiszolgálóra telepítettük (pl. a Mailman weblapokat "https://..." címeken érjük el), akkor annyira biztonságos az azonosítás, amennyire az SSL kapcsolat.

Azonban a legtöbb Mailman telepítés hagyományos, titkosítás nélküli kiszolgálókra történik. Ezzel legtöbbszörre nincs is gond, azonban egy felkészült cracker azonosítás nélkül is képes *lehet* adatokhoz hozzáférni a következő módokon:

* Kapcsolat lehallgatással: A nem nyilvános Mailman lapokon használt azonosításnál a jelszavak sima szöveges formátumban kerülnek elküldésre. Ha ezt el akarjuk kerülni, akkor használjunk SSL-re

képes kiszolgálót.

- * Érvényes süti ellopással: Sikeres bejelentkezés után a Mailman egy sütit küld vissza a felhasználóhoz. A süti a további védett oldalakon segít az azonosításban. A Mailman "kapcsolatig érvényes sütitket" (session cookies) használ, amelyek a böngésző bezárásával, vagy kilépés gombra kattintva lejárnak.

A felhasználó sütijének megszerzésével (pl. a felhasználó böngészőjének sütitket tartalmazó adatbázisának olvasásával, vagy a kapcsolat lehallgatásával, vagy akár olyan hibás böngésző használatával, amely a felhasználó összes sütijét nyilvánossá teszi) és a többi szükséges feltétel egyidejű teljesülésénél jogosulatlan hozzáféréshez lehet jutni.

Fontos tudnunk, hogy ez a módszer könnyebben felhasználható, ha a felhasználó proxy mögött helyezkedik el, mivel ekkor a süti minden az adott proxin átmenő kapcsolatra érvényes lesz, nem csak arra melyet a felhasználó kezdeményezett.

- * Hozzáférés a felhasználó termináljához: Ez is egy süti lopó módszer. Azonban ezt a sütik rövid élettartama nehezíti meg. Fogjuk fel annak, hogy a kényelemért meg kell elégednünk a csökkentett biztonságért, különben minden pillanatban gépelhetjük be a jelszavunkat.

K. A listámról biztonsági másolatot (backup) akarok készíteni. Miket kell lementenem?

A. A válasz a FAQ wizard-ban a következő címen található:

<http://www.python.org/cgi-bin/faqw-mm.py?req=show&file=faq04.006.htm>

K. Hogyan tudom a listát átnevezni?

V. A listák átnevezése jelenleg még elég bonyolult művelet, főleg ha

azt szeretnénk, hogy a régi hivatkozások is éljenek. A jövőben ezen majd változtatni fogunk. :(

A legnagyobb problémát az okozhatja, hogy a lista átnevezése közbeni levélforgalmat hogyan lehet biztonságosan szüneteltetni. Teljesen biztonságos módszer nincsen, de a következőkben leírtak betartásával a hibalehetőséget nagy mértékben lehet csökkenteni:

- Átmenetileg állítsuk le a grunnert. Ehhez a mailman crontab bejegyzését kell módosítani. Adjuk ki a következő parancsot és tegyük minden sor elé megjegyzés jelet ('#'). Mentsük el a változásokat és lépünk ki a szerkesztőből.

```
% crontab -u mailman -e
```

- Állítsuk le a levelezőszerveret. A legtöbb esetben ez nem fog gondot okozni, mivel a távoli MTAK addig próbálkoznak a levél kézbesítésével, amíg a rendszer át nem veszi, s mi nem sok ideig fogjuk a rendszert feltartani.
- Állítsuk le a webszerveret is, ha lehetséges. Ez természetesen azt jelenti, hogy nem lesz elérhető semelyik weboldalunk sem, ezt lehet hogy mégsem szeretnénk. Következő hasznos dolog lesz majd egy állandó átirányítás alkalmazása a régi listáról. Ez azt jelenti, hogyha bárki a régi lista oldalára látogat el, akkor az átirányítás az új listára teszi át. Amíg a lista átnevezésével nem végeztünk addig ez az átirányítás sem fog működni.

Tegyük fel, hogy az "elavult" listát kell átnevezzük "hasznos" listává. Ekkor a következő Apache parancsokat kell használnunk:

```
RedirectMatch permanent /mailman/(.*)/elavult(.*)  
http://www.dom.ain/mailman/$1/hasznos$2
```

```
RedirectMatch permanent /pipermail/elavult(.*)  
http://www.dom.ain/pipermail/hasznos$1
```

Ezeket a sorokat a `httpd.conf` állományba kell elhelyezni, majd indítsuk újra az Apache-ot.

- Ezek után lépünk be a telepített Mailman könyvtárába. Esetünkben legyen ez a `/usr/local/mailman` könyvtár:

```
% cd /usr/local/mailman
```

menjünk a 'lists' könyvtárba:

```
% cd lists
```

Itt találunk egy 'elavult' könyvtárat. Nevezzük át 'hasznos'-sá:

```
% mv elavult hasznos
```

- Most menjünk a privát archívum könyvtárába:

```
% cd ../archives/private
```

Az `elavult.mbox` könyvtárát és a benne található állományokat kell átneveznünk. Most még ne törődjünk a nyilvános archívum idemutató hivatkozásaival, később azokról is gondoskodunk:

```
% mv elavult.mbox hasznos.mbox
```

```
% mv hasznos.mbox/elavult.mbox hasznos.mbox/hasznos.mbox
```

- Most már futtathatjuk a 'bin/move_list' programot az archívum elérési útjainak frissítéséhez. FONTOS: ha Mailman 2.1-et használunk, akkor hagyjuk ki ezt a lépést!

```
% cd ../../
```

```
% bin/move_list hasznos
```


- Hozzuk újra létre a nyilvános archívumot:

```
% bin/arch hasznos
```

- Ezek után néhány lista beállítást is meg kell változtatni, hogyha szeretnénk a régi listára küldött leveleket az új listán látni. Menjünk az új lista adminisztrációs oldalára:

- o Általános beállítások rész

- o A "real_name" résznél adjuk meg a lista új nevét, pl. "Hasznos"

- o Adjuk meg a levél tárgysorába beszúrandó részt (prefix), pl. "[Hasznos] " (igen, fontos a szóköz a végén).

- o Ha szükséges változtassuk meg más beállításokat is, mint például a lista rövid leírása, üdvözlő szövege, stb.

- o Mentsük el a változtatásokat.

- o Privát beállítások rész

- o Adjuk az acceptable_aliases részhez a régi lista címét.

Pl. "elavult@dom.ain" Ezzel (ha a később leírt /etc/aliases módosítást is elvégezzük) a régi listára küldött levelek nem fognak szerkesztői jóváhagyásra várni "a címzett nem egyértelmű" hibával.

- o Mentsük el a változtatásokat.

- Nos, most frissítsük az /etc/aliases állományunkat, hogy fogadja az új lista leveleit és átküldje a régi címre küldött leveleket. A következőkben tárgyaltak Sendmail típusú alias állományoknál működik, eltérő MTA esetén lehet, hogy módosítani kell rajtuk.

- o Keressük meg a régi listához tartozó alias sorokat.
- o Jelöljük ki és másoljuk át közvetlenül a régi alá ezeket a sorokat.
- o Az átmásolt részben minden "elavult" részt írjuk át "hasznos"-sá.
- o Most változtassuk meg a régi listához tartozó címeket, hogy azok az új lista megfelelő címekre mutassanak. Ha mindezt jól csináltuk, akkor a következőhöz hasonlót kell kapnunk:

```
# A régi listát átirányítottuk az új címére
```

```
elavult:          hasznos@dom.ain
elavult-request: hasznos-request@dom.ain
elavult-admin:   hasznos-admin@dom.ain
elavult-owner:   hasznos-owner@dom.ain
```

```
hasznos:         "|/usr/local/mailman/mail/mailman post hasznos"
hasznos-admin:   "|/usr/local/mailman/mail/mailman mailowner hasznos"
hasznos-request: "|/usr/local/mailman/mail/mailman mailcmd hasznos"
hasznos-owner:   hasznos-admin
```

- o Futassuk a 'newaliases' programot.
- Mielőtt mindent újraindítanánk nézzük meg, hogy van-e a qfiles/ könyvtárban a régi listára küldött, de még nem továbbított levél. Ezt a következőképen tehetjük meg:

```
% cd /usr/local/mailman/qfiles
% grep elavult *.msg
```

Ha nincs találat, akkor ugorhatunk a következő lépésre, egy gonddal kevesebb.

Ha van találat, akkor izzadni fogunk egy kicsit. Figyelmeztetlek, hogy a következő lépések nem lettek megfelelően tesztelve. :(

A régi listára küldött minden egyes .msg állománynál a hozzátartozó .db állományt kell módosítani. Sajnos ez nem egy könnyű menet. No lássuk...

Mentsük el a következő Python kód részletet 'hackdb.py' névvel:

```
-----hackdb.py
import sys
import marshal
fp = open(sys.argv[1])
d = marshal.load(fp)
fp.close()
d['listname'] = sys.argv[2]
fp = open(sys.argv[1], 'w')
marshal.dump(d, fp)
fp.close()
-----
```

Adjuk ki azokon az állományokon a következő parancsot, amelyekre a grep találatot jelzett.

```
% python hackdb.py nagyonhosszuhexafilenev1.db hasznos
```

- Ezek után indítsuk el az MTA-t.
- Tegyük újra üzembe a qrunner-t.

```
% crontab -u mailman -e
```

Vegyük ki a megjegyzés jeleket azon sorok elől, ahova mi tettük azokat. Mentsük el a változtatásokat és lépünk ki a szerkesztőből.

- Dőljünk hátra és örüljünk, mert sikerült az átnevezés. Ha 100.000 \$-al támogatód a Mailman fejlesztőgárdáját, akkor ígérjük, hogy hamarosan sokkal könnyebb lesz a listák átnevezése. :)

BSD TANÁCSOK

1. Vivek Khera tanácsa szerint a BSD minden este biztonsági ellenőrzést hajt végre, hogy milyen változások történtek a setuid fájlokon. A setgid könyvtárak is megjelenhetnek az ellenőrzésben, ha azok változtak. Tanácsa szerint BSD esetén nem szükséges a setgid megadása, mivel a könyvtárban létrehozott fájlok automatikusan öröklik a szülők csoport-tulajdonosának jogosultságait. Más Un*xok esetében ez csak akkor történik meg, ha a könyvtárra engedélyezve van a setgid beállítás.

Ha a telepítés során a make parancsnak megadjuk a DIRSETGID változót, akkor a könyvtárak nem kapnak setgid beállítást:

```
% make DIRSETGID=: install
```

Ezzel kikapcsolható a chmod g+s minden egyes könyvtárnál telepítés során.

TITKOSÍTOTT LEVELEZŐLISTÁK

Titkosított levelezőlisták létrehozásához olvassuk el Raphinou
<rb@raphinou.com> leírását a következő címen:

<http://www.raphinou.com/smailman>

AZ EXIM ÉS A MAILMAN EGYÜTT

[Ez a fájl Nigel Metheringhamz "HOGYANIS - Az Exim és a Mailman együtt" dokumentációjának egy kivonata, amelyben a 2.0.x verziójú Mailman-ról és a 3-as verziójú Exim-ről lehet olvasni. A frissített dokumentációban a 2.1-es Mailman és 4-es Exim összekapcsolásáról is lehet olvasni. Az eredeti dokumentációt az alábbi címen lehet olvasni:
<http://www.exim.org/howto/mailman.html>]

Mailman beállítása

A Mailman-t nem kell a hagyományostól eltérő módon beállítani, mint ahogy az a Mailman telepítési dokumentációjában le van írva. Mindössze a mailman uid/gid értékeit kell az Exim konfigurációjának megfelelő helyein megadni.

Exim beállítása

Az Exim beállításával a Mailman-ben létrehozott listák automatikusan, további bejegyzések nélkül megjelennek és használhatóak lesznek.

Ennek a beállításnak a hátránya az, hogy gyenge teljesítménnyel működik olyan rendszereken, amelyek több különböző levelező tartomány listáit szolgálják ki. Igaz, hogy a Mailman képes kezelni virtuális tartományok kezelésére, de egyelőre nem képes kezelni ugyanaz a telepített Mailman azonos névvel, de eltérő tartományban található listákat. Ez a jövőben változni fog. (Lejebb különböző változatokról lehet olvasni, amelyekkel a virtuális tartományokat lehet megfelelően kiszolgálni.)

A később bemutatott konfigurációs állomány részleteket azon már működő Exim konfigurációs részbe kell elhelyezni, amely azon tartomány levelezéséért

felelős, ahol a lista működik. Ha ez a tartomány eltér az Eximben megadott tartományoktól, akkor a következőket kell tenni:

- * adjuk hozzá a local_domains-hez a lista tartományát,
pl. saját.listam.domain
- * adjuk a "domains=sajat.listam.domain" beállítást a lista átirányító (director) részéhez
- * (ha szükséges) a többi átirányítónál vegyük fel ezt a tartományt a figyelmen kívül hagyandó tartományok közé.

[Megjegyzés: a leírásban található beállítások mind 3-as, mind 4-es Eximhez egyaránt használhatóak. 3-as Eximnél 'local_domains' beállításra van szükségünk, míg a 4-es verzióban többnyire a 'local_domains'-ben megadott tartománylistára. Ha nem így használjuk az Eximet, akkor valószínűleg alapos Exim beállítási ismeretekkel rendelkezünk, így nem lesz nehéz módosítani a később ismertetett beállításokat. 4-es Eximben az átirányítók (directors) rész megszűnt, szerepét a routers rész vette át, így ha 4-es Eximet használunk, akkor mindenhol ahol a dokumentáció átirányítót említ mi routert értsünk.]

Függetlenül attól, hogy melyik Exim verziót használjuk az Exim konfigurációs állományának általános beállítások részébe néhány változóz kell megadnunk. Továbbá egy új továbbító részt is létre kell majd hoznunk. 3-as Eximnél egy új átirányító, míg a 4-es verzióban egy új router tölti be majd ugyanazt a szerepet.

Végül pedig érdemes tudni, hogy ezekkel a beállításokkal ugyanazon a gépen egymás mellett működhet a Mailman 2.0-ás és 2.1-es verziója is. Ekkor "mailman" helyett hivatkozhatunk "mm21" névvel a másik, újabb verzióra, pl. MM21_HOME, mm21_transport, stb.

Általános beállítások

~~~~~

Először is ez Exim konfigurációs állományának elejéhez adjuk a következő változókat. A változók segítségével egyszerűbben kezelhetőek a később megadott átirányító és router beállítások. Természetesen a saját rendszerünk szerint módosítsuk a megadott értékeket.

```
# A telepített Mailman 'home' könyvtára
MAILMAN_HOME=/usr/local/mailman
MAILMAN_WRAP=MAILMAN_HOME/mail/mailman
```

```
# Mailman felhasználó és csoport azonosítója, egyeznie kell a Mailman
# beállításakor használt --with-mail-gid kapcsolónál megadott értékkel
MAILMAN_UID=mailman
MAILMAN_GID=mailman
```

Transport rész Exim 3-hoz

~~~~~

Az Exim konfigurációs állományának transports fejezetéhez adjuk a következő sorokat, kb. az első és második "end" tartalmú sor közé kell beszúrni a következőket:

```
mailman_transport:
  driver = pipe
  command = MAILMAN_WRAP \
    '${if def:local_part_suffix \
      ${sg{$local_part_suffix}{-(\\w+)(\\+.*?)}{\$1}} \
      {post}}' \
    $local_part
  current_directory = MAILMAN_HOME
  home_directory = MAILMAN_HOME
  user = MAILMAN_USER
  group = MAILMAN_GROUP
```


Átirányító rész Exim 3-hoz

~~~~~

Ha 3-as Eximet használunk, akkor a konfigurációs állomány átirányító részéhez a következőket kell adni (az átirányítók a második és harmadik "end" tartalmú sor között találhatóak). Fontos, hogy itt számít a sorrend -- pl. ha ezt az irányítót az aliasállomány átirányítójára való hivatkozás elé tesszük, akkor a Mailman bejegyzései (aliases) a rendszer bejegyzéseknél magasabb érvényűek lesznek.

```
# A listához tartozó összes címet kezeli ez a rész.
# Automatikusan azonosítja a listákat a MAILMAN_HOME könyvtárban
# található lists/$local_part/config.pck állományok alapján.
mailman_director:
  driver = smartuser
  require_files = MAILMAN_HOME/lists/$local_part/config.pck
  suffix_optional
  suffix = -bounces : -bounces+* : \
          -confirm+* : -join : -leave : \
          -owner : -request : -admin
  transport = mailman_transport
```

#### Router rész Exim 4-hez

~~~~~

4-es Eximben az átirányítók helyett routereket kell használni. Ezen túl a konfigurációs állomány felépítése is megváltozott -- ABC sorrendet használ, így itt a router rész a transport rész elé kerül. Tegyük a router beállítást a router részbe, amely a "begin routers" sor után kezdődik. El ne felejtsük, hogy a megadott sorrend itt még inkább számít.

```
mailman_router:
```

```
driver = accept
require_files = MAILMAN_HOME/lists/$local_part/config.pck
local_part_suffix_optional
local_part_suffix = -bounces : -bounces+* : \
                    -confirm+* : -join : -leave : \
                    -owner : -request : -admin
transport = mailman_transport
```

Transport rész Exim 4-hez

~~~~~

A 4-es Eximhez használhatjuk a 3-as Eximben megadott transport bejegyzést, nyugodtan másoljuk be valahova a "begin transports" sor után a 3-as Eximnél ismertetett beállításokat.

Megjegyzés

~~~~~

Az Exim-et úgy kell beállítani, hogy a kézbesítés elfogadható mértékű legyen - pl. a max_recipients ne valami nagyon kis érték legyen, és a biztonsági beállítások is értelmes szintet eredményezzenek - pl. 127.0.0.1 címről (evidensen) engedélyezett a relaying, de máshonnan nagyon nem. Párhuzamos küldési módot is be lehet állítani. A kézbesítések időtúllépésére figyelmeztető üzeneteket legalább a listákhoz kapcsolódó címeknél kapcsoljuk ki, ha nem szeretnénk nagyon sok ilyen üzenetet kapni, amikor a címzett gép nem érhető el.

Problémák

~~~~~

- \* A Mailman annyi MAIL FROM/RCPT TO hívást használ, amennyi szükséges. Ennek következtében 10 vagy 100 üzenetnél is többet próbál meg egy

kapcsolat alatt elküldeni, amely több mint az Exim alapesetben beállított engedélyezett küldése (smtp\_accept\_queue\_per\_connection). Emiatt az Exim várakozólista módba kapcsol és nagy késésekkel fogja az üzeneteket továbbítani. A probléma elkerülésére adjunk az SMTP\_MAX\_SESSIONS\_PER\_CONNECTION változónak (a ~mailman/Mailman/mm\_cfg.py állományban) az Exim smtp\_accept\_queue\_per\_connection értékénél kisebb értéket

- \* A Mailman-nak figyelmen kívül kellene hagynia az Exim kézbesítés időtúllépéséből (delay) eredő üzeneteit, annak ellenére, hogy az Exim ezeket sohasem küldi el a listákra. A Mailman 2.1-ben a visszapattanások kezelése és a VERP támogatása nagyban segítheti a visszapattanások nyomonkövetését.
- \* A listák létezésénél ellenőrzi a beállítás, hogy található-e a listához config.pck fájl. Ha töröljük a listát, akkor ügyeljünk hogy töröljük a listához tartozó config.pck fájlt is.
- \* Ha hibás felhasználói azonosítóra hivatkozva nem történik kézbesítés, akkor ellenőrizzük, hogy a MAILMAN\_UID és MAILMAN\_GID változóknak helyesen adtuk-e meg a Mailman ezen értékeit. Figyeljünk arra is, hogy a rendszerszintű aliases fájlban nem található a listához közvetlenül kapcsolódó bejegyzés.

#### Címzett azonosítása

~~~~~

Az Exim címzett azonosítás funkciója nagyon hasznos, mivel segítségével az SMTP kapcsolat ideje alatt el lehet vetni a nem ismert címzettű leveleket. Ez a funkció a saját tartományukba külső címekről érkező levelek esetén hasznos. A Mailman listák forgalmánál a levelek a saját szerverünkről indulnak és különböző külső, tőlünk független tartományokba kerülnek elküldésre. Minden egyes levél több címzettnek kerülhet továbbításra -- akár egyszerre 500-nak is, ha a Mailman alapértelmezett értékét használjuk

és nem módosítottuk az SMTP_MAX_RCPTS értékét.

A Mailman listák forgalmán címzett azonosítást végezve problémák merülhetnek fel. Általában az Exim a Mailman listákról érkező üzenetek minden címzettjét megpróbálja elérni. Ehhez mindössze pár keresést végez a DNSben minden egyes címzettnél, azonban ez a keresés a kézbesítést nagyon lelassíthatja. Ezért célszerű kikapcsolni a címzett azonosítást a Mailman forgalomra.

3-as Exim alatt a következőt kell elhelyezni az általános beállítások részben:

```
receiver_verify_hosts = !127.0.0.1
```

4-es Eximnél valószínűleg az alapértelmezett címzett azonosítási ACL kifejezésben (az "RCPT TO" ACL-ben) már engedélyezve van ez a funkció:

```
accept domains      = +local_domains
endpass
message             = unknown user
verify              = recipient
```

Ekkor csak a saját tartományba érkező levelekre végez címzett azonosítást az Exim. (Ez a megoldás nem teljesen azonos azoknál az üzeneteknél alkalmazott címzett azonosítási funkcióval, amelyek nem-127.0.0.1 hostokról érkeznek, de működik a Mailman esetében.)

SMTP Callback
~~~~~

Az Exim SMTP callback képességével még hatékonyabban lehet azonosítani a hibás feladócímeket, mint a hagyományos feladó ellenőrzéssel. Sajnos nagyon sok szerver a visszapattanási üzenet fejlécében hibás feladót tüntet fel (annak ellenére, hogy csak egy üres feladó címet kéne

használniuk ilyen esetekben).

Azért, hogy a Mailman a visszapattanó címeket kezelni tudja szükséges a Mailman listákon fogadni az ilyen leveleket, még akkor is ha ezek nem alkalmasak visszapattanások azonosítására. Ezért többnyire célszerű kikapcsolni az SMTP callback ellenőrzést a visszapattanó levelekre.

Exim 4 esetén ezt a következő beállítással lehet elérni az "RCPT TO" ACL részben:

```
# Fogadjuk a visszapattanásokat a listán még akkor is ha a callback vagy
# más ellenőrzések nem jártak sikerrel
warn      message      = X-WhitelistedRCPT-nohdrfromcallback: Yes
          condition    = \
            ${if and {{match{$local_part}{(.*)-bounces\+.*}}
                  {exists {MAILMAN_HOME/lists/$1/config.pck}}}} \
              {yes}{no}}
              {yes}{no}}

accept    condition    = \
            ${if and {{match{$local_part}{(.*)-bounces\+.*}}
                  {exists {MAILMAN_HOME/lists/$1/config.pck}}}} \
              {yes}{no}}
              {yes}{no}}

# Most ellenőrizhetjük a feladó címét az SMTP callback funkcióval.
deny      !verify = sender/callout=90s
```

Ha a fejlécben található címekre is szeretnénk SMTP callback ellenőrzést végrehajtani, akkor a "DATA" ACL-t a következőképpen kell megadni:

```
deny      !condition = $header_X-WhitelistedRCPT-nohdrfromcallback:
          !verify = header_sender/callout=90s
```

## VERP használata Eximmel és Mailmannel

~~~~~

VERP esetén a listatagoknak a leveleket külön-külön küldjük el, a VERP beállításáról a `~mailman/Mailman/Defaults.py` állományban lehet bővebben olvasni. Röviden, a következőket kell feltüntetni a `~mailman/Mailman/mm_cfg.py` állományban a VERP használatához:

```
VERP_PASSWORD_REMINDERS = 1
VERP_PERSONALIZED_DELIVERIES = 1
VERP_DELIVERY_INTERVAL = 1
VERP_CONFIRMATIONS = 1
```

(A fenn említett átirányítók képesek a VERP visszapattanások kezelésére is.)

Virtuális tartományok

~~~~~

Virtuális tartományok kezelésére megoldás lehet az a módszer, hogy minden egyes virtuális tartományhoz külön-külön telepített Mailmant használunk. (Jelenleg csak ezzel a módszerrel lehet ugyanazon a rendszeren azonos névvel, de eltérő tartományokban listákat üzemeltetni.)

Ez esetben ne adjuk meg a `MAILMAN_HOME` és `MAILMAN_WRAP` változókat, töröljük azokat az általános beállítások részéből. Módosítsuk a `director` (router) részeket valahogy így:

```
require_files = /virtual/${domain}/mailman/lists/${lc:$local_part}/config.pck
```

a `transports` részeket pedig így:

```
command = /virtual/${domain}/mailman/mail/mailman \  
          ${if def:local_part_suffix \  
            \
```

```
        {{{sg{$local_part_suffix}{-(\\w+)(\\+.*?)?}{\\$1}}}  
        {post}} \  
    $local_part  
current_directory = /virtual/${domain}/mailman  
home_directory = /virtual/${domain}/mailman
```

## Lista ellenőrzése

~~~~~

A következőkben bemutatjuk, hogy hogyan lehet ellenőrizni egy működő rendszeren, hogy az Eximnek megadott listák címei megfelelően működnek. A példában az mems-exchange.org levelezőszerveren működő quixote-users@mems-exchange.org listán mutatjuk be a szükséges parancsokat ("% " jelzi a Unix shell promptját):

```
% exim -bt quixote-users  
quixote-users@mems-exchange.org  
    router = mailman_main_router, transport = mailman_transport  
  
% exim -bt quixote-users-request  
quixote-users-request@mems-exchange.org  
    router = mailman_router, transport = mailman_transport  
  
% exim -bt quixote-users-bounces  
quixote-users-bounces@mems-exchange.org  
    router = mailman_router, transport = mailman_transport  
  
% exim -bt quixote-users-bounces+luser=example.com  
quixote-users-bounces+luser=example.com@mems-exchange.org  
    router = mailman_router, transport = mailman_transport
```

Ha az "exim -bt" parancsok kimenete hasonlít valamennyire a fentiekhez, akkor az jó jel, mert azt jelenti hogy az Exim a megfelelő Mailman bejegyzéshez továbbítja az eltérő üzeneteket. Azonban messze nem jelenti

azt, hogy az Exim és Mailman telepítés biztosan tökéletesen működik!

A dokumentum készítői

~~~~~

Eredeti szerzője Nigel Metheringham <postmaster@exim.org>.

Mailman 2.1-es és Exim 4-es verzióhoz adaptálta Merlin <marc\_soft@merlins.org>.

Ellenőrizte/formába           öntötte/pontosította/egyszerűsítette       Greg       Ward  
<gward@python.net>.



## GNU/LINUX TUDNIVALÓK

Úgy tűnik, hogy a Mailman futtatásához leggyakrabban használt platform a GNU/Linux. A következőkben néhány Linuxos jótanácsot lehet olvasni a Mailman futtatásához:

Ha hibaüzenetet kapunk a kemény kötések (hard link) létrehozásakor, vagy különleges biztonságos kernelt (securelinux/openwall/grsecurity) használunk, akkor olvassuk el a contrib/README.check\_perms\_grsecurity állományt.

Fontos tudni, hogy ugyanez az eset áll fenn, ha a Linux Mandrake-t biztonságos módban (secure mode) használjuk.

Úgy néz ki, hogy a Mandrake 9.0-ban változtattak a gcc jogosultságain, ezért hogy mailman felhasználóként is le tudjuk fordítani a programot, fel kell vennünk a mailman felhasználót a cctools csoportba.

## PYTHON CSOMAGOK

Fontos tudni, hogy ha a Pythont az adott Linux disztribúció csomagkezelőjével (pl. Redhat típusú rendszereknél .rpm-ből, Debian esetén pedig .deb-ből) telepítettük, akkor szükséges telepíteni a Python `fejlesztői' (development) csomagjait is. Különben nem minden fog rendeltetésszerűen működni.

Például: Debian rendszeren Python 2.2 használatakor telepíteni kell a python2.2-dev csomagot is. Redhat rendszeren pedig a python2-devel csomagot.

Ha forrásból telepítjük a Pythont, akkor nem lesz ilyen problémánk.

Az ilyen problémára utal a következő hibaüzenet telepítés során:

```
Traceback (most recent call last):
  File "bin/update", line 44, in ?
    import paths
ImportError: No module named paths
make: *** [update] Error 1
```

Ilyenkor telepítsük a Python fejlesztői csomagot is és próbáljuk meg újra a Mailman telepítését "configure ; make install" paranccsal.

Egyéb Linux disztribúcióknál eltérő módon fordulhat elő ugyanez a probléma, ám bár többnyire a hibaüzenet ImportError formájában jelenik meg.

### **MacOSX TUDNIVALÓK**

A Mailman MacOSX rendszeren is fut, habár a fejlesztők személyesen nem próbálták ezt. A következőkben pár hasznos információt lehet olvasni a Mailman futtatásáról MacOSX-en.

- Jaguar (MacOSX 10.2) a Python 2.2-es verzióját tartalmazza. Habár ez nem a legutolsó stabil Python verzió a Mailman 2.1 működéséhez megfelel.
- David B. O'Donnell leírása Mailman 2.0.13 és Postfix használatáról MacOSX kiszolgálón az alábbi címen olvasható:

<http://www.afp548.com/Articles/mail/python-mailman.html>

- Kathleen Webb tanácsai a Mailman futtatásáról Jaguar rendszeren Sendmail-lel a következő címen tekinthetők meg:

<http://mail.python.org/pipermail/mailman-users/2002-October/022944.html>

## NETSCAPE PROBLÉMA

Azon felhasználóknál, akik member-only listára Netscape Communicatorral próbálnak meg levelet küldeni, problémák merülhetnek fel. Megfigyelhető, hogy Linuxon a Communicator 4.x hibás Sender: fejléct szűr a küldendő levelekbe -- pl. Sender: fejlécbe mindössze a feladó e-mail címének felhasználói név része kerül. A Netscape többi verziójánál is előfordulhat ez a hiba.

Alapesetben a members-only listák először a From: fejléc alapján próbálják meg azonosítani a beküldőt, ha ez sikertelen akkor a Sender: résszel próbálkoznak. A Mailman-t adminisztrátora beállíthatja, hogy először mindig a Sender: fejléct használja fel azonosításhoz. Ilyenkor, ha a Sender: rész hibásan van kitöltve, akkor nem fog egyezni egyetlen listatag címével sem, a levél ezért a jóváhagyásra váró levelek közé kerül.

A jövőben a Mailman fejlettebb algoritmussal próbálja majd azonosítani a beküldőket, de addig M. A. Lemburg <mal@lemburg.com> tanácsa segít a probléma megoldásában. Érdeemes elküldeni ezt a részletet azon felhasználóknak, akik levele ok nélkül kerül a jóváhagyásra váró levelek közé.

A .netscape könyvtárban található két .js állományba (liprefs.js és preferences.js) kell elhelyezni az alábbi sort:

```
user_pref("mail.suppress_sender_header", true);
```

A bináris állományban található megjegyzés szerint ezt a sort csak Unix rendszereken kell beszúrni.

Mivel a Communicator minden kilépéskor újra létrehozza ezt az állományt,

ezért előbb lépünk ki a Communicatorból, és csak akkor módosítsuk az állományokat ha már nem fut a program. A következő indítás után már

nem fogja a Sender: fejléctet beszúrni, így minden újra jól fog működni.

Az ok amiért minden újra jól fog működni az, hogy a Mailman a From: fejléctet fogja újra az azonosításhoz használni, ha nem talál Sender: fejléctet, még akkor is ha a rendszeradminisztrátor először Sender: fejléc ellenőrzést adott meg.

#### MOZILLA

A Mozilla 0.9.x verzióknál ilyen probléma még nem merült fel.

## **BIZTONSÁGI FIGYELMEZTETÉSEK**

Semmiképp se csábítson a DELIVERY\_MODULE változó `Sendmail'-re való beállítása az mm\_cfg.py állományban, ha a rendszered Sendmail-t használ MTA-nak. A Sendmail.py modul neve becsapós, mert valójában egy parancssori továbbítási metódust jelent szemben az (alapértelmezett) SMTPDirect SMTP továbbítási metódusával. Sendmail.py használata biztonsági kockázatot jelent, szerepe mindössze egy tartalék módszer. Ha valamiért nem működik az SMTPDirect.py, akkor a biztonsági kockázatok elkerülése érdekében inkább orvosuljuk a problémát és ne használjuk a Sendmail.py-t, mert ezzel biztonsági rést nyithatunk a rendszerünkön.

## SENDMAIL `smrsh' KOMPATIBILITÁS

A legtöbb újabb verziójú Sendmail egy úgynevezett korlátozott végrehajtású eszközzel az "smrsh"-val rendelkezik. Ennek a használatával lehet korlátozni, hogy a Sendmail milyen levélszűrő programokat hívhat meg működésekor. Külön meg kell adnunk az smrsh -nak a Mailman wrapper programját, különben a Mailman nem fog működni. Ha a levél nem jut el a Mailman wrapper programjához, vagy egy "operating system error" üzenetet látunk a mail naplóállományba, akkor a problémát ez okozza.

A probléma elhárításához tegyük a következőt:

- keressük meg a rendszerünkön a Sendmailhez tartozó smrsh wrapper programot

```
% grep smrsh /etc/mail/sendmail.cf
```

- derítsük ki, hogy az smrsh mely könyvtárakban keresi az engedélyezett szűrő programokat. A következő parancs

kimenetének legelején találhatóak meg teljes hivatkozással  
a lehetséges könyvtárak, pl. /var/admin/sm.bin

```
% strings $path_to_smrsh | less
```

- lépünk be az így megtalált könyvtárba, pl. /var/adm/sm.bin

```
% cd /var/adm/sm.bin
```

- hozzunk létre egy szimbolikus hivatkozást a Mailman wrapper  
programjára. Példánkban a Mailman az alapértelmezett könyv-  
tárba telepítettük (/usr/local/mailman)

```
% ln -s /usr/local/mailman/mail/mailman mailman
```

## SENDMAIL ÉS MAILMAN ÖSSZEKAPCSOLÁSA

David Champion készített egy leírást arról, hogy hogyan lehet a  
Mailman-t és a Sendmail-t úgy összekapcsolni, hogy automatikusan  
felismerje és használja a Sendmail a frissen létrehozott  
levelezőlistákat anélkül, hogy minden egyes alkalommal kézzel  
kéne módosítani a bejegyzéseket.

A contrib könyvtárból az alábbi négy állományra lesz szükség:

|                   |                                                     |
|-------------------|-----------------------------------------------------|
| mm-handler.readme | - részletes leírás a szükséges beállításokról       |
| mm-handler        | - a levéltovábbító program (MDA)                    |
| mailman.mc        | - egy minta konfigurációs állomány                  |
| virtusertable     | - minta a kivételként kezelendő címekkel (RFC 2142) |

## TUDNIVALÓK A TELJESÍTMÉNYRŐL

Az egyik legbiztosabb módszer a Sendmail teljesítményének javítására

az ha kikapcsoljuk a címzettek folyamatos DNS-beli azonosítását. Ha nem tesszük ezt meg, akkor a Mailman nagyon lassan fogja fel- dolgozni a leveleket. A Mailman általában a localhost-on (pl. 127.0.0.1) keresztül kapcsolódik a Sendmail SMTP portjához, ilyenkor célszerű a Sendmail úgy beállítani, hogy ne végezzen folyamatosan DNS ellenőrzést a localhost csatlakozásoknál.



## KOMPATIBILITÁS

Az RFC 2369-ben foglaltak szerint minden listakezelő-programnak List-\* fejléceket kell adniuk minden egyes kimenő levélhez. Ezek a fejlécek a levelezőprogramokat (MUA) készítik fel, hogy segítsék a felhasználókat az olyan levelezőlistákkal kapcsolatos feladatokban, mint információ lekérdezése vagy leiratkozás. A Mailman az RFC-ben foglaltak alapján hozza létre ezeket a fejléceket.

Jelenleg nem mindegyik MUA képes az RFC 2369 fejlécek értelmezésére, így ahelyett hogy azokat elrejtene, megmutatják a felhasználóknak. Sok listaüzemeltetőt kérdeznek felhasználói ezekről a fejlécekről.

A 2.0-ás Mailman-ben a forrás módosítása nélkül nem lehet a List-\* fejléceket eltüntetni, mivel ezek hosszútávon a felhasználók kényelmét szolgálják. A fejléceket ne távolítsuk el, inkább a MUA készítőjét kérjük meg a fejlécek támogatására. A változtatásig a következő tanácsokkal segíthetünk felhasználóinknak e fejlécek eltüntetésében.

### EUDORA FELHASZNÁLÓKNÁL

Mike Noyes tanácsa szerint:

Az új lista fejléceket a következő módon lehet elrejtetni. Nyissuk meg az Eudora.ini állományt, majd a következő sort adjuk hozzá a [settings] részhez:

```
TabooHeaders=List,X-UID,Received,Status,X-UIDL,Message,In-Reply, \  
X-Priority,Mime-Version,Content,X-Persona,Resent-Message,References, \  
Return,X400,X-400,Mail-System,Errors-To,X-List,Delivery,Disposition, \  
X-Juno,Precedence,X-Attachments,X-MSMail,X-MimeOLE,X-Nav
```

megjegyzés: az alapértelmezés minden ami nem "List" tartalmú

Hivatkozás Eudora .ini Settings TabooHeaders

<http://www.eudora.com/techsupport/ini.html>