Postfix és MariaDB/MySQL konfigurálása

Fontos, hogy a blackPanther OS-ben előre definiált konfigurációs segítik a szolgáltatások beüzemelését. A dokumentációban található leírás inkább iránymutatás mint kötelező értékek. Természetesen a szükséges alapbeállításoktól nem térhetünk el, de a Postfix jól értelmezhető naplózással rendelkezik, könnyedén megértjük a hibaüzeneteket.

Az idő múlásával egyes programok beállításai megváltozhattak, egyes programok helyett újak, jobbak jöttek. Kérlek ne felejtsd el, hogy a szabad szoftverek világában állandó a változás!

Az SMTP az angol „Simple Mail Transfer Protocol” kifejezés rövidítése. Ez egy kommunikációs protokoll az e-mailek Interneten történő továbbítására. Az SMTP egy viszonylag egyszerű, szöveg alapú protokoll, ahol egy üzenetnek egy vagy több címzettje is lehet.

Könnyen tesztelhetjük az SMTP-t a „telnet” program segítségével. Az SMTP szolgáltatás a TCP (Transmission Control Protocol) 25-ös portját használja. Ahhoz, hogy meghatározza, hogy az adott tartomány névhez melyik SMTP szerver tartozik, a tartomány név MX (Mail eXchanger) rekordját használja. Ez a tartomány DNS rekordjai között szerepel.

A szerver FQDN nevét el kell helyezni az alábbi fájlban.

```
mcedit /etc/mailname
mail.magyarparduc.hu
```

Nyissuk meg szerkesztésre a postfix main.cf nevű fájlját.

```
mcedit /etc/postfix/main.cf
```

Adjuk meg az internet hostnevet a gépünknek.

```
myhostname= mail.magyarparduc.hu
```

Írjuk át a tartomány nevünkre a következőt, mely meghatározza azt a tartomány nevet,ahonnan a helyileg elküldött levél megérkezni látszik és ahova a helyileg küldött levél megérkezik.

```
# myorigin=/etc/mailname
myorigin=$myhostname
```

Lehetőség van megadni az SMTP üdvözlő üzenetet.

```
smtpd_banner = $myhostname ESMTP $mail_name (blackPanther OS/GNU)
```

Az alábbit hagyjuk üresen, hiszen nem továbbítunk másik szervernek leveleket közvetlenül rendszer szinten, helyben kezeljük a leveleket.

```
relayhost =
```

Definiáljuk, mely hálózati interfészen figyeljen, várjon kapcsolódást a Postfix.

```
inet_interfaces = all
```

Csak localhost-ról, azaz a szerverről engedélyezzük a kapcsolódást a Postfix-hez, azaz más SMTP klienst nem engedünk közvetlenül csatlakozni.

```
mynetworks_style = host
```

Adjuk hozzá a helyi kézbesítéshez szükséges alias táblák elérhetőségeit. Amennyiben ezen
táblákat módosítjuk, a „ newaliases ” paranccsal aktiválhatjuk a változásokat.

```
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
local_recipient_maps = unix:passwd.byname $alias_database
```

A fentiek alapján csinálni kell egy álneveket tartalmazó fájlt. Ez csupán helyi használatra kell.

```
cp /etc/aliases /etc/postfix/aliases
```

```
postalias /etc/postfix/aliases
```

Adjuk meg a helyi tartománynevet. Fontos, hogy ne adjunk meg a virtuálisan definiált tartománynevek közül egyet se.

```
mydestination = $myhostname
```

Az alap Postfix konfiguráció nem tartalmaz semmilyen fajta spam-szűrést, de azt hiszem nem is lenne célszerű, mivel nem lehet tudni, hogy telepítés után a Postfix-et mire fogják használni (intranet, mail-hub, relay, stb...). Ezért az adminisztrátornak a körülményekre való tekintettel kell konfigurálnia az anti-UCE szabályait.

Postfix-ben három fajta korlátozás van:

1. korlátozási szakasz
2. korlátozás
3. hozzáférhetőségi névsor

A Postfix a korlátozási szakaszokat a következő sorrendben dolgozza fel:

```
smtpd_client_restrictions
smtpd_helo_restrictions
smtpd_sender_restrictions
smtpd_recipient_restrictions
smtpd_data_restrictions
```

Ezek rövid magyarázata:

smtpd_client_restricions – Az MTA-k egymás között szoktak egyfajta „beszélgetés”-t folytatni, eközben az egyik a kliens a másik a szerver. Mindkettő ismeri az SMTP protokollt. Ez a fajta korlátozási szakasz az SMTP kliens címét és host nevét korlátozza. A kliens ebben az esetben az MTA-hoz kapcsolódó TCP/IP klienst jelenti.

smtpd_helo_restrictions – Általában az smtp kliensek küldenek az SMTP szerverhez egy HELO/EHLO -t, amivel elárulják, hogy ők milyen host névről kapcsolódnak a szerverhez. Ezzel a szakasszal megadhatjuk, hogy milyen host nevet küldhetnek az SMTP kliensek (például nem lenne célszerű saját host nevüket küldeniük, azzal kiadva, hogy az smtp-kliens=smtp-szerver).

Célszerű beállítani, hogy egy SMTP kliensnek küldenie kell a 19HELO/EHLO-t, ezzel sok UCE (Unsolicited commercial e-mail – kéretlen reklám levél) küldő programot megszűrhetünk.

Ezt így tehetjük meg:

```
smtpd_helo_required=yes
```

smtpd_sender_restrictions – Ezzel a szakasszal beállíthatjuk, hogy milyen küldőt fogadjon el a Postfix MTA a „MAIL FROM:” parancsban.

Mivel sok spammer, azaz kéretlen levélküldő a „MAIL FROM:”-hoz nem létező vagy érvénytelen e-mail címet ír (mint például valaki@foobar.hu vagy valaki@foo@bar.hu). Itt persze feltételezzük, hogy a foobar.hu
domain nem létezik.

smtpd_recipient_restrictions – Ez a korlátozási szakasz szabályozza, hogy az SMTP kliens mit küldhet a „RCPT TO:” parancsnak paraméterként. Az úgynevezett spammerek természetesen

itt is használhatnak érvénytelen e-mail címet, amiket különböző opciókkal
kiszűrhetünk.

smtpd_data_restrictions – Ez magát a „DATA” parancsot korlátozza.

Kiszűri azt a fajta SMTP klienst, mely nem az SMTP szabályai szerint működik. Nagyon fontos, hogy a különböző korlátozásokat jó sorrendbe rakjuk, mert aszerint dönti el a Postfix, hogy továbbadja-e a processzt vagy megállítja.

Ennek fényében a következőket állítjuk be:

```
       smtpd_helo_required = yes
       smtpd_helo_restrictions = permit_mynetworks, warn_if_reject
       reject_non_fqdn_hostname, reject_invalid_hostname, permit
       smtpd_sender_restrictions = permit_mynetworks, warn_if_reject
       reject_non_fqdn_sender, reject_unknown_sender_domain,
       reject_unauth_pipelining, permit
       smtpd_client_restrictions = reject_rbl_client sbl.spamhaus.org,
       reject_rbl_client blackholes.easynet.nl, reject_rbl_client
dnsbl.njabl.org
       smtpd_recipient_restrictions = reject_unauth_pipelining,
permit_mynetworks,
       reject_non_fqdn_recipient, reject_unknown_recipient_domain,
       reject_unauth_destination, permit smtpd_data_restrictions =
       reject_unauth_pipelining
```

Több tartomány kezeléséhez szükséges paraméterek következnek az alábbiakban.

Adjuk meg a levelek tárolásának helyét a fájlrendszerben *(ez az alapértelmezett)*.

```
virtual_mailbox_base = /var/spool/mail
```

Felhasználói fiókok helyét tartalmazó adatbázisbeli tábla helyét leíró fájl megadása:

```
virtual_mailbox_maps =
```

Amennyiben használni akarjuk a MySQL kiterjesztést akkor ezeket kell beállítanunk és ugorj most a 16. oldalra mert létre kell hoznod az adatbázist is:

```
virtual_mailbox_base = /var/spool/mail/virtual
virtual_mailbox_maps = mysql:/etc/postfix/mysql_mailbox.cf
```

Felhasználói azonosítót tartalmazó adatbázisbeli tábla helyét leíró fájl megadása.

```
virtual_uid_maps = mysql:/etc/postfix/mysql_uid.cf
```

Felhasználói csoport azonosítót tartalmazó adatbázisbeli tábla helyét leíró fájl megadása.

```
virtual_gid_maps = mysql:/etc/postfix/mysql_gid.cf
```

Álneveket tartalmazó adatbázisbeli tábla helyét leíró fájl megadása.

```
virtual_alias_maps = mysql:/etc/postfix/mysql_alias.cf
```

Levelezési tartományokat tartalmazó adatbázisbeli tábla helyét leíró fájl megadása.

```
virtual_mailbox_domains = mysql:/etc/postfix/mysql_domains.cf
```

Hozzuk létre a levelek tárolását szolgáló mappát, továbbá a tulajdonosát is:

```
mkdir -p /var/spool/mail/virtual
groupadd virtual -g 5000
useradd virtual -u 5000 -g 5000
chown -R virtual:virtual /var/spool/mail/virtual
```

Hozzuk létre a fentiekben hivatkozott MySQL adatbázis csatlakozáshoz szükséges fájlokat, melyekben megadjuk a csatlakozáshoz szükséges felhasználói adatokat, úgymint felhasználói név és hozzá a jelszó, az adattábla nevét, a lekérdezni szánt adatot tartalmazó mezőt, az adatbázis-kezelőt futtató szerver IP címét és a lekérdezéshez szükséges egyéb feltételeket, amennyiben szükséges. Először a felhasználói postafiókok meghatározásához szükségeset.

```
mcedit /etc/postfix/mysql_mailbox.cf
```

```
user=mail
password=qwer1234
dbname=maildb
table=users
select_field=maildir
where_field=id
hosts=127.0.0.1
additional_conditions = and enabled = 1
```

Majd a felhasználói azonosítókhoz szükségeset.

```
mcedit /etc/postfix/mysql_uid.cf
```

```
user=mail
password=qwer1234
dbname=maildb
table=users
select_field=uid
21where_field=id
hosts=127.0.0.1
```

A csoportazonosító meghatározásához.

```
mcedit /etc/postfix/mysql_gid.cf
```

```
user=mail
password=qwer1234
dbname=maildb
table=users
select_field=gid
where_field=id
hosts=127.0.0.1
```

Az álnevek táblához valót.

```
mcedit /etc/postfix/mysql_alias.cf
```

```
user=mail
password=qwer1234
dbname=maildb
table=aliases
select_field=destination
where_field=mail
hosts=127.0.0.1
additional_conditions = and enabled = 1
```

A domainek táblához.

```
mcedit /etc/postfix/mysql_domains.cf
```

```
user=mail
password=qwer1234
dbname=maildb
table=domains
select_field=domain
where_field=domain
hosts=127.0.0.1
additional_conditions = and enabled = 1
```

Courier beállítása

/a csomag neve: courier-authdaemon/

Most állítsuk be a Courier-t, hogy MySQL adatbázisból végezze a felhasználók azonosítását és kapcsoljuk be a naplózást ideiglenesen a tesztelési fázis végéig.

Ehhez szerkesszük a „ /etc/courier/authdaemonrc ” fájlt, hogy az „ authmodulelist ” a következő legyen:

```
mcedit /etc/courier/authdaemonrc
```

```
authmodulelist="authmysql"
DEBUG_LOGIN=2
```

Állítsuk be a hitelesítéshez szükséges paramétereket a „/etc/courier/authmysqlrc ” fájlban.

```
mcedit /etc/courier/authmysqlrc
```

```
MYSQL_USERNAME mail
MYSQL_PASSWORD qwer1234
MYSQL_DATABASE maildb
MYSQL_USER_TABLE users
MYSQL_CRYPT_PWFIELD crypt
# MYSQL_CLEAR_PWFIELD clear
```

Adatbetöltés:
Először a rendszerszintű, azaz helyi levelezés kezeléséhez szükséges adatokat helyezzük el az adatbázisba.

```
mysql -u mail -p maildb
```

```
INSERT INTO domains (domain) VALUES ('localhost'),
('localhost.localdomain');

INSERT INTO aliases (mail,destination) VALUES
('postmaster@localhost','root@localhost'),
('sysadmin@localhost','root@localhost'),
('webmaster@localhost','root@localhost'),
('abuse@localhost','root@localhost'),
('root@localhost','root@localhost'),
('@localhost','root@localhost'),
('@localhost.localdomain','@localhost');

INSERT INTO users (id,name,maildir,crypt) VALUES
('root@localhost','root','root/', encrypt('qwer1234') );
```

Majd az általunk kezelt tartományokat.

```
INSERT INTO domains (domain) VALUES ('magyarparduc.hu');
INSERT INTO aliases (mail,destination) VALUES ('@magyarparduc.hu',
'postmaster@magyarparduc.hu'),                                    ('postmaster
magyarparduc.hu','postmaster@magyarparduc.hu'),
('abuse@magyarparduc.hu','postmaster@magyarparduc.hu');
```

Az „aliases” táblában lehetőség van megadni, hogy adott címeket miképp kezeljen a rendszer. Értékpárokat adunk meg. Az első a címzettet jelöli, a második érték pedig hogy hova kerüljön kézbesítésre a levél. Megadunk egy azonosító nélküli értéket is „@hostnév.tld” formában. Ez arra az esetre lehet jó, ha kezelni szeretnénk azokat a beérkező leveleket is amelyeknek a címzettjei nincsenek explicit megadva a címlistában, de szeretnénk az összes általunk kezelt tartományra érkező levelet kezelni. Ez a „catch-all” módszer jól jöhet kis számú levélforgalmat és/vagy kevés címzettet kezelő helyeken, azonban nagyobb számú levél esetén vagy vállalati környezetben kényelmetlenné vagy szinte használhatatlanná válhat.

Hátrányaként sorolható fel, hogy a feladó nem értesül a rossz vagy helytelen címzésről. Általam nem támogatott technológia, ezért csak mint lehetőséget mutattam be.

A táblában ha a „destination” mezőben olyan címet adunk meg, melyet nem mi kezelünk, akkor a levél csak továbbításra kerül az adott külső címre, különben a rendszer kézbesíti a helyi felhasználó levélmappájába.

```
INSERT INTO users (id,name,maildir,crypt) VALUES
('postmaster@magyarparduc.hu','Postmaster','postmaster/',
encrypt('qwer1234') );
```

Mint látható, a felhasználók megadásánál négy értéket adok meg. Az „ id ”-t, ami a felhasználó e-mail címe lesz és egyben azonosítja is, a „ name ” mezőt, mely a nevét takarja, a „ maildir ”-t, mely a „/var/spool/mail/virtual/” könyvtáron belüli helyét adja meg. Fontos, hogy „/”-re végződjön, különben nem használható mint UNIX könyvtár forma. A „crypt” pedig a jelszót tartalmazza.

Ekkorra egy kész és működőképes, használatba vehető rendszerünk van. Gyakran próbálunk úgy felépíteni egy rendszert, hogy mindent azonnal feltelepítünk és csak azután teszteljük le a működését. Így nehézkesebb megtalálni az esetlegesen előforduló hibákat, melyek lehet csak tényleg egy-egy karakternyi elgépelésből adódnak. Éppen ezért mielőtt továbbmennénk, érdemes megvizsgálni a jelen rendszert.

Tartalom szűrők beüzemelése

/csomagnevek: amavisd-new spamassasin spamassasin-spamd clamav postgrey/

Az Amavisd-new beállításához szükséges állományok a „/etc/amavisd” könyvtárban találhatóak.

Tegyük megjegyzéssé a vírus és spam figyelését egyelőre.

```
mcedit /etc/amavis/15-content_filter_mode
```

```
# @bypass_virus_checks_maps = (
# \%bypass_virus_checks, \@bypass_virus_checks_acl, \
$bypass_virus_checks_re);
# @bypass_spam_checks_maps = (
# \%bypass_spam_checks, \@bypass_spam_checks_acl, \
$bypass_spam_checks_re);
```

Helyezzük el a következő pár sort a „/etc/amavis/50-user” fájlban, mellyel definiáljuk a naplózási szintet, azt az értéket, melytől egy levél spamnek minősül, továbbá a spamek sorsát (továbbításra kerül).

```
mcedit /etc/amavis/50-user
```

```
@local_domains_acl = qw(.);
$log_level = 2;
$syslog_priority = 'debug';
$sa_kill_level_deflt = 8.0;
$final_spam_destiny = D_PASS;
```

Állítsuk be a Postfix-et, hogy kommunikáljon az Amavis-el. Tegyük a Postfix „/etc/postfix/master.cf” konfigurációs fájljának végére a következő sorokat, mellyel megadjuk milyen tartalomszűrő módszereket használjon, továbbá definiáljuk a kommunikációt közöttük. A következő rész a Postfix kimenő Amavis-al SMTP-n létesített kapcsolataira vonatkozik.

A „/etc/postfix/main.cf” fájl „content_filter = amavis:

[127.0.0.1]:10024” sora tartozik ide, mellyel megadjuk a Postfix-nek, hogy minden a „/etc/postfix/master.cf” fájlban megadott Amavis felületen keresztül minden bejövő levelet küldjön ki a 127.0.0.1 címre a10024-es TCP porton, az Amavis alapértelmezett SMTP figyelő kapuján át.

```
amavis unix - - - - 2 smtp
-o smtp_data_done_timeout=1200
-o smtp_send_xforward_command=yes
-o disable_dns_lookups=yes -o max_use=20
```

Ez a részlet adja meg azt a bejövő felületet, melyen a Postfix-nek fogadnia kell az Amavis által visszaadott üzeneteket. A Postfix tehát a 127.0.0.1-es IP címen a 10025-ös alapértelmezett porton figyeli az Amavis által küldött értesítéseket és üzeneteket.

```
127.0.0.1:10025 inet n - - - - smtpd
-o content_filter=
-o local_recipient_maps=
-o relay_recipient_maps=
-o smtpd_restriction_classes=
-o smtpd_delay_reject=no
-o smtpd_client_restrictions=permit_mynetworks,reject
-o smtpd_helo_restrictions=
-o smtpd_sender_restrictions=
-o smtpd_recipient_restrictions=permit_mynetworks,reject
-o smtpd_data_restrictions=reject_unauth_pipelining
-o smtpd_end_of_data_restrictions=
-o mynetworks=127.0.0.0/8
-o smtpd_error_sleep_time=0
-o smtpd_soft_error_limit=1001
-o smtpd_hard_error_limit=1000
-o smtpd_client_connection_count_limit=0
-o smtpd_client_connection_rate_limit=0
-o receive_override_options=no_header_body_checks,
no_unknown_recipient_checks
```

Adjuk hozzá az átvevő transzport részhez a következő két sort, hogy kikapcsoljuk a fejrész és levéltörzs figyelést, ezt majd az Amavis fogja elvégezni.

```
-o content_filter=
-o receive_override_options=no_header_body_checks
```

Adjuk hozzá a kapcsolódási pontot a „ /etc/postfix/main.cf ” állományban.

```
mcedit /etc/postfix/main.cf
```

```
content_filter = amavis:[127.0.0.1]:10024
```

Adjuk hozzá a „clamav” rendszerfelhasználót „amavis” csoporttagsággal, hogy legyen jogosultsága az Amavis átmeneti állományait vizsgálni.

```
adduser clamav amavis
```

Az alapértelmezett Spamassassin beállításokat fogjuk használni a Bayes módszerrel kiegészítve. Engedélyezzük a szűrő indulását a rendszerrel.

```
szolgaltatasok spamd be
```

```
mcedit /etc/spamassassin/local.rf
```

```
use_bayes 1
bayes_path /etc/spamassassin/bayes
bayes_file_mode 0770
```

Ha összegyűlt legalább kétszáz darab spam és nem spam levelünk, lehetőségünk van Bayes szűrőt tanítani vele az alábbi módon az „sa-learn” paranccsal.

```
sa-learn --showdots -C /etc/spamassassin --spam \
/var/spool/mail/virtual/quarantine/.spam/*

sa-learn --showdots -C /etc/spamassassin --ham
/var/spool/mail/virtual/mine/cur/*
```

Érdemes a későbbiekben is tanítani figyelve arra, hogy mindkét típusból adagoljunk neki elegendő mennyiséget, így elkerülhetjük a hibás minősítéseket.

Ezzel beállítottuk az Amavis-t a Postfix-hez. Tesztelhetjük a levélküldést. Ha mindent rendben találunk, akkor vegyük ki a megjegyzést a korábban elhelyezett „/etc/amavis/15-content-filter-mode” állományban.

```
mcedit /etc/amavis/15-content-filter-mode
```

```
@bypass_virus_checks_maps = (
\%bypass_virus_checks, \@bypass_virus_checks_acl, \
$bypass_virus_checks_re);
@bypass_spam_checks_maps = ( \%bypass_spam_checks,
\@bypass_spam_checks_acl, \$bypass_spam_checks_re);
```

A naplózást és a spam levelek továbbítását pedig kapcsoljuk ki az alábbi módon.

```
vim /etc/amavis/conf.d/50-user
@local_domains_acl = qw(.);
$log_level = 1;
$syslog_priority = 'info';
$sa_kill_level_deflt = 8.0;
$final_spam_destiny = D_DISCARD;
```

Hitelesítés

Az eddigi konfiguráció esetén a kliens és szerver közötti kommunikáció úgynevezett nem rejtjelezett szövegként történik. Ez lehetőséget nyújt illetékteleneknek jelszavunk vagy akár teljes levelezésünk eltulajdonítására a kommunikációs csatorna lehallgatásával. Ezt elkerülendő titkosítjuk mind a jelszavak átvitelét, mind a teljes kommunikációt a levelező kliens és szerver között.

A SASL a jelszó titkosításával biztosítja az adott azonosítást, ezáltal a jelszavakat nem lehet könnyen megszerezni. Engedélyezzük a Postfix-nek a SASL fájljaihoz való hozzáférést.

```
adduser postfix sasl
```

Mivel a Postfix biztonsági okokból alapértelmezetten „chroot”-olt környezetben fut, biztosítanunk kell számára az „sasl” démon elérhetőségét.
Némi magyarázat a „chroot” megértésehez. A „chroot” futtatja a parancsot vagy az interaktív shell-t a paraméterben megadott speciális gyökérkönyvtár értékkel. Haszna az, hogy miután „chroot”-oltunk egy speciálisan felkészített mappába, onnantól kezdve az aktuális héjon belül, a gyökérkönyvtárnak a rendszer az adott mappát, és nem a valódit tekinti. Ezzel egy kisebb rendszert hozhatunk létre a nagyobbik rendszeren belül.

Gyakran jelen van Linux telepítőkészletekben, és úgynevezett LiveCD-k használatakor.

```
mkdir -p /var/spool/postfix/var/run/saslauthd
```

A Postfix-hez adjuk hozzá a SASL konfigurációt.

```
mcedit  /etc/postfix/main.cf
```

```
# SASL
smtpd_sasl_auth_enable = yes
broken_sasl_auth_clients = no
smtpd_sasl_security_options = noanonymous
smtpd_sasl_local_domain =
```

A beállításokat módosítsuk, hogy az „smtpd_sender_restrictions”-höz és az „smtpd_recipient_restrictions”-höz hozzáadjuk a „**permit_sasl_authenticated**” értéket.

```
smtpd_sender_restrictions = permit_sasl_authenticated,
permit_mynetworks,
```

```
warn_if_reject reject_non_fqdn_sender, reject_unknown_sender_domain,
reject_unauth_pipelining, permit
smtpd_recipient_restrictions = reject_unauth_pipelining,
permit_mynetworks,
permit_sasl_authenticated, reject_non_fqdn_recipient,
reject_unknown_recipient_domain, reject_unauth_destination,
check_policy_service inet:127.0.0.1:10023, permit
```

A SASLAUTHD indítását és futtatását állítsuk be úgy, hogy a Postfix sorban fusson.

```
START=yes
OPTIONS="-r -c -m /var/spool/postfix/var/run/saslauthd"
```

Hozzuk létre a következő fájlt ha nem létezik és állítsuk be a Postfix-nek, hogyan használja a SASL-t.

```
mcedit /etc/postfix/sasl/smtpd.conf
```

```
pwcheck_method: saslauthd
mech_list: plain login cram-md5 digest-md5
log_level: 7
allow_plaintext: true
auxprop_plugin: mysql
sql_engine: mysql
sql_hostnames: 127.0.0.1
sql_user: mail
sql_passw : qwer1234
sql_database: maildb sql_select: select crypt from users where
id='%u@%r'
and enabled = 1
```

A naplózást ki is kapcsolhatjuk, miután meggyőződtünk a helyes működéséről.Mondjuk meg a PAM-nak, miképp hitelesítse az SMTP-t MySQL-en keresztül.

```
mcedit /etc/pam.d/smtp
```

```
auth required pam_mysql.so user=mail passwd=qwer1234 host=127.0.0.1
db=maildb table=users usercolumn=id passwdcolumn=crypt crypt=1
account sufficient pam_mysql.so user=mail passwd=qwer1234 host=127.0.0.1
db=maildb table=users usercolumn=id passwdcolumn=crypt crypt=1
```

A beállítások végén indítsuk újra a szolgáltatásokat és teszteljük le a működését levélküldéssel.

```
szolgaltatasok saslauthd ujrainditas
Szolgaltatasok postfix ujrainditas
```

A Courier-ben is állítsuk be a SASL-t. Ez a sor megjegyzésként van a konfigurációs fájlban, nézzük meg, hogy egyforma legyen, vegyük ki a megjegyzést, majd indítsuk újra a szolgáltatásokat és tesztelhetjük le működésüket.

```
mcedit /etc/courier/imapd
```

```
IMAP_CAPABILITY="IMAP4rev1 UIDPLUS CHILDREN NAMESPACE THREAD=ORDEREDSUBJECT
THREAD=REFERENCES SORT QUOTA AUTH=CRAM-MD5 AUTH=CRAM-SHA1 IDLE"
```

Indítsuk újra a szolgáltatásokat.

```
szolgaltatasok courier-authdaemon ujrainditas
szolgaltatasok courier-imap ujrainditas
```

Az újabb változatokban a courier-imap helyét a cyrus-imap program vette át!

Titkosítás

Ahhoz, hogy a szerver és a kliensek közötti kommunikáció is titkosítva legyen, a Courier-t és a Postfix-et kell megfelelő módon bekonfigurálni. A levelek olvasásához a Courier-ben kell a titkosítást beállítani, míg küldéshez a Postfix-ben.

A titkosításhoz szükség lesz tanúsítványok elkészítésére.

```
openssl req -new -outform PEM \ -out postfix.cert -newkey rsa:2048
-nodes
-keyout postfix.key -keyform PEM -days 999 -x509
openssl req -x509 -newkey rsa:1024 -keyout imapd.pem -out imapd.pem
-nodes
-days 999
```

Mindkét esetben meg kell adni a szükséges információkat, mint az ország, állam vagy tartomány neve, lokalizáció nevét – úgymint város például, szervezet nevét, egy nevet – mint például a létrehozó nevét e-mail címet.

```
Country Name (2 letter code) [AU]:HU
State or Province Name (full name) [Some-State]:Hajdu-Bihar
Locality Name (eg, city) []:Debrecen
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Példa cég
Organizational Unit Name (eg, section) []:Példa cég
Common Name (eg, YOUR name) []:Létrehozó neve
Email Address []:valaki@magyarparduc.hu
```

Kezdjük a Postfix beállításával. Néhány TLS beállítás már adott a konfigurációban, nézzük át,
hogy az alábbiak szerepelnek-e a következő paraméterekkel.

```
mcedit /etc/postfix/main.cf
```

```
#smtp_use_tls = no
```

```
smtp_tls_security_level = may
30#smtpd_use_tls=yes
smtpd_tls_security_level = may
#smtpd_tls_auth_only = no
smtp_tls_note_starttls_offer = yes
smtpd_tls_loglevel = 1
smtpd_tls_received_header = yes
smtpd_tls_session_cache_timeout = 3600s
tls_random_source = dev:/dev/urandom
smtpd_tls_cert_file=/etc/ssl/certs/postfix.pem
smtpd_tls_key_file=/etc/ssl/private/postfix.key
#smtpd_tls_session_cache_database = btree:$
{data_directory}/smtpd_scache
#smtp_tls_session_cache_database = btree:${data_directory}/smtp_scache
```

A „/etc/postfix/master.cf” fájlban is nézzük át, majd állítsuk át hasonló módon az alábbiakat.

```
mcedit /etc/postfix/master.cf
```

```
submission inet n - n - - smtpd
-o smtpd_sasl_auth_enable=yes
-o smtpd_tls_auth_only=yes
# -o smtpd_tls_security_level=encrypt
# -o header_checks=
# -o body_checks=
-o smtpd_client_restrictions=permit_sasl_authenticated,
reject_unauth_destination,reject
-o smtpd_sasl_security_options=noanonymous,noplaintext
-o smtpd_sasl_tls_security_options=noanonymous
# -o milter_macro_daemon_name=ORIGINATING smtps inet n - - - - smtpd
-o smtpd_tls_wrappermode=yes
-o smtpd_sasl_auth_enable=yes
-o smtpd_tls_auth_only=yes
-o smtpd_client_restrictions=permit_sasl_authenticated,reject
-o smtpd_sasl_security_options=noanonymous,noplaintext
-o smtpd_sasl_tls_security_options=noanonymous #
-o milter_macro_daemon_name=ORIGINATING
```

A fenti beállításokkal csak a TLS titkosításos autentikáció lesz lehetséges. A tanúsítvány fájlokat másoljuk a konfigurációban megadott helyekre. Következzen a Courier beállítása. Adjuk meg a Courier-nek, hogy hol éri el fájlszinten a tanúsítványokat.

```
mcedit /etc/courier/imapd-ssl
```

```
TLS_CERTFILE=/etc/courier/imapd.pem
```

A MySQL beállítása

Elsőként létre kell hozni a kapcsolódáshoz egy felhasználót akinek a nevében történnek majd az adatbázis műveletek. Létre kell hoznunk továbbá a táblákat, melyek tartalmazni fogják a rendszer által kezelt tartományokat, a hozzájuk tartozó felhasználókat és egyéb rendszerspecifikus paramétereket.

Lépjünk be a MySQL CLI felületére a „root” felhasználóval és annak jelszavával:

```
mysql -u root -p
```

Hozzuk létre az adatbázist:

```
create database maildb;
```

Hozzuk létre a fent említett „mail” nevű felhasználót a megfelelő jogosultságokkal, azaz legyen joga lekérdezésre, módosításra, adattábla és adat törlésére, adattábla és adat létrehozására. Itt kell megadnunk a felhasználó jelszavát is.

```
GRANT SELECT,INSERT,UPDATE,DELETE,CREATE,DROP ON maildb.* TO
'mail'@'localhost' IDENTIFIED by 'qwer1234';
GRANT   SELECT,INSERT,UPDATE,DELETE,CREATE,DROP   ON   maildb.*   TO
'mail'@'%'
IDENTIFIED by 'qwer1234';
exit;
```

Lépjünk ki majd be az újonnan létrehozott felhasználónkkal:

```
mysql -u mail -p maildb
```

Hozzuk létre az „aliases”, „domains” és „users” táblákat.

```
CREATE TABLE `aliases` (
`pkid` smallint(3) NOT NULL auto_increment,
`mail` varchar(120) NOT NULL default '',
`destination` varchar(120) NOT NULL default '',
`enabled` tinyint(1) NOT NULL default '1',
PRIMARY KEY (`pkid`),
UNIQUE KEY `mail` (`mail`) ) ;
CREATE TABLE `domains` (
`pkid` smallint(6) NOT NULL auto_increment,
`domain` varchar(120) NOT NULL default '',
`transport` varchar(120) NOT NULL default 'virtual:',
`enabled` tinyint(1) NOT NULL default '1',
16PRIMARY KEY (`pkid`) ) ;
CREATE TABLE `users` (
`id` varchar(128) NOT NULL default '',
`name` varchar(128) NOT NULL default '',
```

```
`uid` smallint(5) unsigned NOT NULL default '5000',
`gid` smallint(5) unsigned NOT NULL default '5000',
`home` varchar(255) NOT NULL default '/var/spool/mail/virtual',
`maildir` varchar(255) NOT NULL default 'akarmi/',
`enabled` tinyint(3) unsigned NOT NULL default '1',
`change_password` tinyint(3) unsigned NOT NULL default '1',
`clear` varchar(128) NOT NULL default 'ChangeMe',
`crypt` varchar(128) NOT NULL default 'sdtf21h4dxj66',
`quota` varchar(255) NOT NULL default '',
`procmailrc` varchar(128) NOT NULL default '',
`spamassassinrc` varchar(128) NOT NULL default '',
PRIMARY KEY (`id`),
UNIQUE KEY `id` (`id`) ) ;
exit;
```

Az „alias” tábla, azaz álnév tábla olyan névpárosokat fog tartalmazni, ahol az első értékkel egy valós, létező postafiókra hivatkozunk. Így lehetőség van egy címhez több nevet használni.

A „domains” táblában tároljuk a rendszer által kezelt tartományneveket, míg a „users” táblában ezen tartományokhoz tartozó postafiókokat. Egyelőre kapcsoljuk be a MySQL-ben a naplózást, amit a tesztelés után kikapcsolhatunk, lévén eléggé teljesítmény igényes. Szedjük ki a megjegyzést.

```
mcedit /etc/mysql/my.cnf
```

log = /var/log/mysql/mysql.log

Érvénybe léptetéséhez újra kell indítani a szolgáltatást:

```
Szolgaltatasok mysqld ujrainditas
```

phpMyAdmin

A phpMyAdmin egy könnyen kezelhető grafikus felületet nyújt a MySQL adminisztrációjához. Egy böngészőn keresztül tudunk módosításokat eszközölni az adatbázisban, nem szükséges parancssori hozzáférésen keresztül parancssorból megtenni azt. Apache webkiszolgáló és PHP modul futtatása szükséges hozzá. Nagymértékű rálátást biztosít a meglévő adatbázisunkról. Telepítése:

```
telepites phpmyadmin
```