SZAKDOLGOZAT

Sztancs László

Debrecen

2010

Debreceni Egyetem

Informatikai Kar

Komplex levelező rendszer kialakítása Linux környezetben

Témavezető:

Ecsedi Kornél

Készítette:

Sztancs László

Programozó

matematikus

Debrecen

2010

Tartalomjegyzék

1 Bevezetés.....1
2 Telepítés.....2
2.1 Az alap rendszer.....2
2.2 Forráslista.....2
2.3 Csomagok.....3
2.4 Adatbázis-kezelő.....4
2.5 MTA.....6
2.6 Adatbiztonság.....9
2.7 IMAP kiszolgáló.....10
2.8 Tartalomszűrők.....11
2.9 Webmail.....13
2.10 phpMyAdmin.....14
2.11 Tűzfal.....14
3 Konfigurálás.....15
3.1 A MySQL beállítása.....16
3.2 Postfix konfigurálása.....17
3.3 Courier beállítása.....22
3.4 Adatfeltöltés.....23
3.5 Tartalom szűrők beüzemelése.....25
3.6 Autentikáció.....28
3.7 Titkosítás.....30
3.8 Webmail.....32
3.9 Tűzfal.....33
4 Tesztelés.....36
5 Összefoglalás, továbbfejlesztési lehetőségek.....43
6 Köszönetnyilvánítás.....44
7 Rövidítések jegyzéke.....45
8 Mellékletek.....47
9 Irodalomjegyzék.....61

1 Bevezetés

Jelen írással szeretnék egy összefogó képet adni mindazoknak, akik most készülnek saját levelező rendszert kialakítani otthoni vagy céges használatra. Az alább bemutatásra kerülő produktum egyaránt alkalmas hobbi szintű, kis-, közép- és nagyvállalati környezetben való elhelyezésre. Ez a sokrétű felhasználhatósága a skálázhatóságából, sebességéből, stabilitásából és konzisztenciájából, továbbá a felhasznált csomagok készleten lévőségéből (stock packages) adódik.

Utóbbi fontosságát kiemelném. Nem én vagyok az első és biztos nem az utolsó aki megörököl egy már élesben működő – legyen az levelező vagy bármilyen más szolgáltatást nyújtó – rendszert. Sajnos sok esetben elmarad a telepítésükre és üzemeltetésükre vonatkozó dokumentálás, így üzemeltetésük, napra készen tartásuk sok esetben nehézkessé, vagy egyenesen lehetetlenné válik. Nincs mindig lehetőség kész csomagokkal dolgozni, de jelen írásban szereplő rendszer felépítése megkötések nélkül elkészíthető e módon. Ezáltal azonnal két dologgal is megkönnyítjük életünket: nincs szükség egyedi csomagok elkészítésére és a frissítés is megtörténhet komolyabb nem tervezett leállás nélkül. Mindez könnyebbséget nyújt azoknak is akik nincsenek olyan tudásnak birtokában amihez csak több éves megtapasztalás útján juthat el az ember vagy csak szeretik az azonnal használható – out of the box – jellegű megoldásokat kompromisszumok nélkül.

A felhasználói levelek elérhetőségét POP3, IMAP és HTTP alkalmazási protokollokon, továbbá ezek SSL feletti biztonsági változatain keresztül biztosítjuk. Ide kapcsolódik szorosan, hogy telepítésre kerül egy nyílt forráskódú Apache HTTP webkiszolgáló is, mely nem csak a levelek elérését hivatott kiszolgálni, hanem bizonyos szűkített adminisztrációs célokat is elvégezhetünk segítségével.

A szakdolgozatban ismertetésre kerülnek a telepítés lépései, kitérve a fontosabb részekre, helyet adva a testre szabhatóságnak, ügyelve az Internet-szabványok betartására, továbbá bemutatom a teszteléshez, üzembe helyezéshez és üzemeltetéshez szükséges feltételeket és eszközöket.

2 Telepítés

2.1 Az alap rendszer

Alapját a Debian GNU/Linux szabad (free)[1] operációs rendszer szolgáltatja, mely Linux kernelt[2] használ, a rajta futó operációs rendszereszközök többsége pedig a GNU projektből[3] származik (innen ered a GNU/Linux név).

Természetesen van lehetőség más disztribúció választására, amennyiben a telepítendő csomagok elérhetőek, azonban erre jelen dolgozatban nem térek ki. A megvalósítandó rendszer 32 vagy 64 bites x86 architektúrán kerül telepítésre.

2.2 Forráslista

Az APT az „Advanced Package Tool” angol kifejezés rövidítése, azaz egy fejlett csomagkezelő eszközről van szó. Az APT három fő részből áll: az apt-get és apt-cache programból, és a source.list – azaz a beállítási – fájlból (avagy forrásfájlból). A source.list fájl olyan HTTP és FTP címeket tartalmaz, ahonnan az APT adatokat kérhet le. Így a forrásainkat egyetlen fájlban tudjuk csoportosítani. Egy bejegyzés a következőképpen néz ki:

```
deb http://ftp.hu.debian.org/debian/ lenny main contrib non-free
```

Az első mező lehet „`deb`” vagy „`deb-src`”. Előbbi, ha olyan sorról van szó, mely végrehajtható állományok helyét határozza meg. Utóbbi azt mondja meg, hogy honnan szedhetünk le Debian forráskód csomagokat. A következő az URI, mely a Debian könyvtárszerkezet gyökerét határozza meg. Az ezt követő határozza meg, mely disztribúció csomagjait vehetjük innen. Ez lehet nevesített, azaz például „`lenny`”, vagy lehet a fejlesztési állapotára vonatkozó, mint a „`stable`”, „`unstable`”. Az alkotóelemek általában „`main`”, „`contrib`” „`non-us/contrib`”, „`non-us/main`” névre hallgatnak.

A fájl szerkesztése után le kell futtatnunk az „**apt-get update**” parancsot, ezzel

aktualizálva azt a rendszer felé.

Az „**apt-get**” a Debian csomagkezelő eszköze, mely kezeli a csomagok közötti függőségeket és ellentmondásokat. Csomag telepítéséhez az „**apt-get install <csomagnév>**” formában van lehetőségünk. Amennyiben más csomagokat is kell telepíteni a csomagunkhoz, azt az APT jelezni fogja, mint ahogy az ellentmondásokat is, azaz amiket el kell távolítani, mert ütközést okoznának. Ha a rendszeren található összes csomagot szeretnénk frissíteni, akkor egyetlen paranccsal megtehetjük: **apt-get upgrade**. Ha olyan csomagot telepítettünk, melyet mégis szeretnénk eltávolítani, az „**apt-get remove <csomagnév>**”-el tehetjük meg. Teljes Linux-disztribúció frissítést pedig az „**apt-get dist-upgrade**” paranccsal tudunk eszközölni.

Csomagokról adatokat az „**apt-cache**” eszköz segítségével kérhetünk le telepítés nélkül. Az „**apt-cache show <csomagnév>**” megjeleníti ezen adatokat, míg az „**apt-cache depends <csomagnév>**” listát ad arról, hogy milyen más csomagok szükségesek az adott csomag telepítéséhez.

A rendszert a következő forráslista segítségével telepítjük:

/etc/apt/sources.list

```
deb     http://ftp.hu.debian.org/debian/ lenny         main contrib non-
free
deb-src http://ftp.hu.debian.org/debian/ lenny         main contrib non-
free
deb     http://security.debian.org/      lenny/updates main
deb-src http://security.debian.org/      lenny/updates main
```

2.3 Csomagok

Minden csomag egy tárállomány, mely „**.deb**” kiterjesztéssel végződik. A Postfix MTA-t tartalmazó csomag neve például ez: **postfix_2.5.5-1.1_i386.deb**. A **postfix** a szoftver neve, a **2.5.5** tag az adott szoftver verziószámát, az **1.1** a csomagváltozatot jelöli, az **i386** pedig az architektúrára utal.

Először a rendszer napra készre állítását kell elvégezni:

```
apt-get update
```

```
apt-get upgrade
```

Ezzel a rendszer alap telepítésekor felkerült csomagokat frissítjük naprakészre.

2.4 Adatbázis-kezelő

A felhasználók adatait többféleképpen tárolhatjuk a rendszerben. Rendszer szintű fájlokban, adatbázisban vagy címtár szolgáltatásban, azaz LDAP-ban. Én adatbázisban fogom tárolni ezen adatokat a könnyebb kezelhetőség végett. A MySQL és PosgreSQL két ingyenes, nyílt forráskódú adatbázis-kezelő. Nekem a MySQL-re esett a választásom a könnyű adminisztrálhatóság, egyszerűbb telepíthetőség (például a „full-text index” benne van az alaptelepítésben) és a kevesebb funkció miatt (nincs idegen kulcs MyISAM típusú tábláknál) jóval gyorsabb egyszerűbb műveleteknél.

Tények, jelek, betűk, számok összességét nevezzük adatnak, melyet a számítógép dolgoz fel, hogy értelmezhető adatot állítson elő belőle. Számítógépes rendszerekben ezen elemeket többnyire állományokban tárolják. Az egymással összefüggő állományok pedig adatbázis alkotnak. Sorba és oszlopokba rendezzük az adatokat a fájlokban. A relációs adatbázis-kezelők ezen adatokat sorokból és oszlopokból álló táblázatba rendezik. A felhasználó által feltett kérdésekre pedig a táblázatok között létrehozott relációk segítségével adható válasz. Ezeket a kérdéseket nevezzük lekérdezéseknek. A relációs adatbázis-kezelő választ ad minden lekérdezésre, függetlenül attól, hogy az adatokat frissítjük, lekérdezzük, avagy töröljük.

A MySQL eredeti fejlesztője a svéd MySQL AB cég, amely kettős licenceléssel tette elérhetővé a MySQL-t; választható módon vagy a GPL, vagy egy kereskedelmi licenc érvényes a felhasználásra. 2008 januárjában a Sun felvásárolta 800 millió dollárért a céget. 2010. január 27-én a Sun-t felvásárolta az Oracle Corporation, így a MySQL is Oracle tulajdonba került.

Az egyik legelterjedtebb adatbázis-kezelő, aminek egyik oka lehet, hogy a teljesen nyílt forráskódú LAMP (Linux–Apache–MySQL–PHP) összeállítás részeként költséghatékony és egyszerűen beállítható megoldást ad dinamikus webhelyek szolgáltatására.[4]

A MySQL 5.x képességei:

•ANSI SQL 99, számos kiegészítéssel

•Keresztplatformos elérhetőség

•Tárolt eljárások

•Adatbázis triggerek

•Kurzor adatbázisok

•„View” adatbázisok

•Valódi VARCHAR támogatás

•INFORMATION_SCHEMA támogatás

•„Strict” (szigorú) mód

•X/Open XA elosztott tranzakció-feldolgozás (DTP) támogatása; az Innobase InnoDB motorjának használata

•Különálló tároló motorok,(MyISAM olvasási sebességért, InnoDB a tranzakciókhoz és a referenciális integrációhoz, MySQL Archive az elavult adatok kevés helyen történő tárolására

•Tranzakciók az InnoDB, BDB és Cluster tároló motorokkal

•SSL támogatás

•Lekérdezés gyorstár (cache)

•Egymásba ágyazott SELECT -ek

•Szöveges indexelés és keresés a MyISAM motorral

•Beágyazott adatbázis-könyvtár

•Részleges Unicode támogatás

•ACID megfelelés az InnoDB-vel, BDB-vel és Cluster-rel

•Továbbfejlesztett MySQL Cluster

•„Példányosítás”

A következő képességekkel a MySQL rendelkezik, számos más relációs adatbázisrendszerrel ellentétben.

•Többféle tároló motor, amelyek között bármely táblához szabadon választhatunk

•Natív tároló motorok (MyISAM, Falcon, Merge, Memory (heap), MySQL Federated, MySQL Archive, CSV, Blackhole, MySQL Cluster, Berkeley DB, EXAMPLE, és Maria)

•Partnerek által fejlesztett tároló motorok (InnoDB, solidDB, NitroEDB, BrightHouse)

•Közösségi fejlesztésű tároló motorok (memcached, httpd, PBXT, Revision Engine)

•Akár egyéni tároló motor

•Commitek csoportosítása, több tranzakció fogadása többféle kapcsolatról, melyek meggyorsítják a tranzakciók lefolyását.

Nagyon gyors, többfelhasználós, többszálú, erőteljes SQL adatbázis-kiszolgáló. 1996-ban jelent meg a piacon első kiadása forráskóddal együtt. Bizonyítottan villámgyors és megbízható, melyet egyre több cég használ a világon. Nagyméretű rekordhalmazok kezelésénél is rendkívül gyors.[5]

Az adatbázis-kezelőt a következőképpen telepítjük:

```
apt-get install mysql-server mysql-client
```

Telepítéskor kérni fogja a „`root`” felhasználó jelszavát, melyet majd meg kell adni.

2.5 MTA

Az MTA rövidítés jelentése: „Mail Transfer Agent”, magyarul levéltovábbító ügynök. [6] Az internetes levélkezelő rendszereken belül, a transzferáló ügynök egy olyan

számítógépes eljárás vagy szoftveres ügynök ami egyik gépről a másikra továbbítja az elektronikus leveleket, egyetlen „ugrással”, alkalmazás-szintű tranzakciókban. Egy ilyen MTA magában foglalja az SMTP-nek a kliens (küldés) és a szerver (fogadás) részét is. A levelező szerver fogalmát viszont sokszor olyan gépekre is értik, amik MTA-ként viselkednek, a megfelelő szoftvert futtatva. A „mail exchanger” (MX a DNS szövegkörnyezetében) formálisan egy olyan domain névre utal, ami hozzá van rendelve egy eszközhöz, ami levelező szerverként működik, kiterjesztett értelemben pedig magát a szervert jelenti.[7]

Sokféle MTA telepíthető a rendszerre, mint a Sendmail, Exim (alapértelmezettként települ Debianra) vagy a Qmail. Sok pro és kontra érv szól egyik és másik mellett, melyek eredménye általában az, hogy mindig a felhasználási célnak megfelelőt válasszuk. Nálam a döntésnél kulcsszerepet játszott a könnyű telepíthetőség, a sebesség, a soros és párhuzamos terhelhetőség (egyik levél elmegy és csak utána a következő vagy párhuzamosan indulnak a levelek) és a biztonság. Rendszeremhez a Postfix-et választottam, mert moduláris felépítésű, könnyen kezelhető, gyors, jól támogatott.

A Postfix egy MTA, mely szabadon elérhető az IBM Public License alatt. Eredetileg Wietse Venema kezdte el fejleszteni az IBM támogatásával Vmailer név alatt, de később nevet kellett változtatni, mivel az említett nevet már más termék használta, így született a Postfix név. Méltán híres teljesítményéről, biztonságosságáról, és igen széles körű konfigurálási lehetőségeiről, ideértve pl. akár LDAP alapú leképezéseket, és különböző spam/UCE szűrési lehetőségeket is. A Postfix fejlesztésénél fő szempont továbbá a „szép” kód, és a kompatibilitás más MTA-kkal, mely jelenti az RFC-k pontos betartását, vagy akár a Sendmail-ből, illetve Qmail-ből (például: „`maildir`” támogatás) ismert adminisztrálási megoldásokkal való kompatibilitást, lehetővé téve a könnyű átállást Postfix-re.[8]

Felépítését tekintve moduláris, minden szolgáltatást külön programrész hajt végre, a jogosultságok pedig a lehetséges legalacsonyabb szintre kerültek. A rendszert a „`main.cf`” és a „`master.cf`” két fő beállítófájlon keresztül tudjuk módosítani. Úgy tervezték meg, hogy a lehető legkevesebb modul érintkezzen a távoli hálózatokkal, azok pedig a legkisebb jogosultsági szinttel rendelkezzenek.

A levelek továbbításáért felelős főbb programok/modulok az alábbiak:

smtpd – ez fogadja az internetről vagy a belső hálózatról érkező leveleket és különböző

ellenőrzések után átadja azokat a „cleanup” programnak. Tud vizsgálatot végezni, úgymint a küldő fél open-relay - azaz nyitott továbbító-e avagy sem -, nem tiltottuk-e ki a feladót más módon, például tartomány alapján. De akár itt engedélyezhetjük azt is, hogy olyantól érkezzen levél, aki valamilyen más módon egyébként tiltva van.

cleanup – ellenőrzést végez, hogy a levelek fejléce megfelel-e az előírásoknak (például a címzett valós létező cím). Vizsgálja, hogy kell-e valamilyen módosítást végeznie a fejlécen, esetleg meg kell-e hívni a „trivial-rewrite”-ot. Ha végezett, akkor fájlba rakja a levelet, elhelyezni a bejövő sorba. Ebbe a sorba kerülnek a rendszer által elfogadott levelek.

trivial-rewrite – elvégzi a levél fejlécének módosítását majd visszaadja a „cleanup”-nak. A programot a „qmgr”, azaz a sorkezelő is meg tudja hívni.

qmgr – a bejövő sorból áthelyezi a levelet az aktív sorba, továbbá felülegyeli a levél sorsát. Az aktív sorból kerülnek a levelek továbbításra helyi postafiókba vagy távolra. A „qmgr” figyeli az aktív sor telítettségét, nehogy túlterhelődjön a kiszolgáló vagy el ne fogyjon a memória és ezáltal a többi levél kezelését se tudja ellátni. Arra is figyel, hogy azon levelek, melyeket először próbál elküldeni, előnyt élvezzenek azokkal szemben, melyeket nem sikerült először elküldeni és ezért későbbi továbbításra várnak. A levelek innen ahhoz a programhoz kerülnek, melyet a továbbítás megkövetel, azaz a „local”, „smtp” vagy a „pipe” programhoz.

local – fogadj a leveleket a helyi felhasználók számára, ellenőrzi az alias adatbázist és .forward fájl meglétét. A „.forward” fájlon keresztül a vezérlést más programoknak tudjuk átadni vagy megadhatjuk benne, ha egy másik e-mail címre akarjuk továbbítani a leveleket. Az alias adatbázist arra használjuk, hogy neveket képezzünk le más nevekre. Így egy névhez akár több név is rendelhető, vagy a rendszer által ismert névre mutathatunk.

smtp – továbbítja a leveleket nem helyi felhasználóknak. Tehát ez a program biztosítja a kapcsolatot a távoli levélkiszolgálóval. Kizárólag az SMTP protokollon keresztül tartja a kapcsolatot a távoli rendszerekkel.

pipe – más eltérő rendszernek továbbítja helyi szinten a leveleket, például egy adatbázisba.

sendmail – ez a program gyakorlatilag egy burkoló, helyettesíti a Sendmail-t magát, az ahhoz megírt programok helyi kéréseit átfordítja a Postfix által értelmezhető hívásokká, és leteszi azokat az előírt könyvtárba. Ha a könyvtár mindenki által írható, akkor ezt egyedül is

meg tudja tenni, ha azonban csak egy adott csoport számára elérhető, mely általában a „maildrop” csoport, akkor a „postdrop” programot hívja meg erre a célra.

pickup – a „maildrop” könyvtárból átadja a leveleket a „cleanup” programnak.

Telepítsük a Postfix-et MySQL támogatással.

```
apt-get install postfix postfix-mysql
```

Ez eltávolítja az Exim alapértelmezésként felkerült levelező motort és telepíti többek között az openssl és ssl-cert csomagokat.

Meg kell adni a levelező szerver típusát. 6 opció választható, melyek az alábbiak:

„No configuration”, „Internet site”, „Internet with smarthost”, „Satellite system”, „Local only”. Esetünkben az „Internet site” kerül kiválasztásra, azaz közvetlenül itt történik a levelek kezelése SMTP használatával.

A levelező szerver teljes domain nevét (FQDN) is itt kell megadni (telepítéskor ajánl egy alapértelmezettet, melyet korrigálhatunk). A Fully Qualified Domain Name = Hibátlan Teljes Domain-Nevet (FQDN) szokták abszolút/teljes tartomány névnek is hívni, ami egy olyan tartomány név, ami teljesen pontosan meghatározza a helyet a Domain Name System = Tartomány Név Rendszer (DNS) fa hierarchiájában. [9]

2.6 Adatbiztonság

Ahhoz, hogy rendszerünk ne legyen kéretlen levélküldők áldozata és elkerüljük ennek következményeit, mindenképpen korlátoznunk kell azon helyek számát, amelyek átjáróként használhatják szerverünket. Erre ad megoldást a hitelesített SMTP vagy más néven az SMTP auth. Lényege, hogy miután a távoli gép IP-címe folyamatosan változhat, felhasználónévhez és jelszóhoz kötjük a csatlakozást; amennyiben ez érvényes, úgy a kliens levélküldő szerverként használhatja gépünket.

A SASL a „Simple Authentication Layer” angol kifejezés rövidítése, egy azonosításra és adatbiztonságra szolgáló keretrendszer, és a 2222-es számú RFC taglalja részletesen.

Segítségével egyszerű és hatékony autentikációs felületet biztosíthatunk egyes alkalmazásaink számára. A telepítésre kerülő csomagok a Cyrus SASL API implementációi SQL kiterjesztéssel, melyek támogatják a MySQL-t, továbbá egy PERL interfész a SASL-hoz, és az a modul, mely lehetővé teszi a MySQL-ből való azonosítást.

```
apt-get install libsasl2-modules-sql libgsasl7 libauthen-sasl-cyrus-perl
sasl2-bin libpam-mysql
```

2.7 IMAP kiszolgáló

Többféle mailbox-szerver érhető el Debian alatt. Ilyen a Courier, a Cyrus és a Dovecot. A Cyrus fürtözhető, aktívan fejlesztik, nagyon gyors, lévén saját postaláda-formátumot használ, de emiatt nehézkes visszaállítani hiba esetén a leveleket. A Dovecot szintén „Maildir" postaláda-formátumot használ akárcsak a Courier, sebességük is hasonló, aktívan fejlesztik szintén, biztonságos, kezeli a SASL-t, viszont tapasztalataim miatt a kiegyensúlyozott tulajdonságokat figyelembe véve a Courier-t választottam.

A Courier a szokványos IMAP kiszolgálóktól annyiban tér el, hogy csak a „Maildir" levéltárolási formát és a sajátját, azaz a „Maildir++"-t támogatja. A „Maildir" esetén a beérkező levelek külön állományban kerülnek tárolásra a felhasználó saját könyvtára alatt található „`Maildir`" mappában. Az új levelek ennek megfelelően a „`~Maildir/new/`", míg az olvasottak a „~/Maildir/cur" alatt vannak. Van még egy átmeneti mappa is, a „`~Maildir/tmp`", melyet a rendszer ideiglenesen tárolásra használ mozgatások idejére, mint a „`/new`"-ból a „`/cur`" könyvtárba való áthelyezéskor vagy amíg biztonságosan el nem mentette a beérkező levelet a „/new" könyvtárba. A „Maildir++" abban különbözik az eredeti formától, hogy támogatja a tárkorlátot, azaz a quota-t. A quota segítségével megadható, hogy adott felhasználó mekkora tárterülettel gazdálkodhat vagy hány fájlt tárolhat. Másik nagyon fontos tulajdonsága, hogy szinte bármilyen felhasználó azonosításra képes, többek között PAM-on keresztül MySQL-ből. Mindemellett akár többféle azonosítást is használhatunk egyszerre. Támogatja továbbá az SSL-t. A Courier-IMAP kitűnően együttműködik a Postfix kiszolgálóval.[10]

Az alapcsomagokat, a MySQL és a SSL támogatást nyújtó csomagokat a következőképpen telepítjük:

```
apt-get install courier-base courier-authdaemon courier-authlib-mysql
courier-imap courier-imap-ssl courier-ssl
```

telepítéskor webkönyvtárak létrehozását ajánlja fel a „`/etc/courier`” könyvtárban konfigurációs fájl létrehozása helyett, melyeknek a Courier-webadmin program használatakor lenne jelentősége. A courier-ssl csomag X.509-es SSL tanúsítvány elkészítésére figyelmeztet, melyet a courier-imap-ssl csomag telepítésekor automatikusan elvégez.

Az SSL (Secure Socket Layer) egy protokoll réteg, amely a szállítási rétegbeli protokoll (pl. TCP/IP) és valamely alkalmazási rétegbeli protokoll (pl. HTTP) között helyezkedik el, az OSI terminológia szerinti viszony- és megjelenítési réteg feladatait látva el. Webböngészésnél például az SSL biztosítja a biztonságos kommunikációt a kliens (böngésző) és a szerver (webszerver) között. Autentikációhoz digitálisan aláírt tanúsítványokat használ, a kommunikáció titkosítva zajlik (az SSL handshake során közösen megegyeznek egy kulcsban, ebből generálják azután az egy session erejéig használatos session keyt, és ezt használják valamely szimmetrikus titkosító algoritmussal, például DES, AES, stb.).[11]

2.8 Tartalomszűrők

A számítógépes vírusok elsöprő többsége elektronikus levelezés útján terjed. Ezeknek és a kéretlen, azaz spam leveleknek a szűrését érdemes már a gyökereknél megszűrni, azaz a levelező kiszolgálón.

A spam olyan elektronikus levél, mely számunkra nem hordoz tartalmat, feleslegesen foglalja le erőforrásainkat (a mi és a gép energiáit egyaránt), továbbá sokszor félrevezető információkat hordoz.[12]

Debian alá fellelhető ingyenes alkalmazásokat keresve találtam rá az Amavis-Clamav-SpamAssasin nyílt forráskódú hármasra, melyek együttműködve nagy hatékonysággal képesek megszűrni a levelezésünkre, gépünkre ártalmas leveleket.

Az Amavisd-new egy levelező-szűrő. Fogadja a leveleket, majd azokról megállapítja,

hogy tartalmaznak-e vírusokat vagy spameket, majd ennek megfelelően elutasítja, karanténba helyezi vagy továbbítja azokat a levelező kiszolgáló felé. Több, mint 30 népszerű víruskeresőt támogat, köztük a ClamAV-ot is. Amennyiben több víruskeresőnk van, azokat csoportokba sorolhatjuk arra az esetre, ha az egyik nem lenne elérhető akkor tudjon a másikhoz fordulni. Nagy forgalmú helyeken az Amavisd-new-t és a beépülő SpamAssasin-t és ClamAV-ot külön tartalomszűrő gépre telepítik vagy ilyen gépek terheléselosztásos rendszerét építik fel. PERL nyelven íródott a biztonságot és megbízhatóságot szem előtt tartva. Magja az RFC-nek megfelelő levélkezelő, melyet úgy terveztek, hogy egy levelet se veszítsen el, azaz mindaddig nem vesz át levelet, amíg az alatta elhelyezkedő levelező kiszolgáló át nem veszi azt. Négyféle szűrési lehetőséget kínál: spam szűrés, vírus/malware szűrés, veszélyes mellékletek és érvénytelen fejlécek tiltása. Többféle vírusirtót képes meghívni, esetünkben ez a nyílt forráskódú ClamAV lesz, a spamszűrést pedig a beépülő SpamAssassinnal végzi.[13]

A SpamAssassin sok spamszűrő módszerrel rendelkezik. Ilyen a Bayes-féle tanuló módszer, az SPF keresése, képes jellegfelismerésre. A levelek tesztelésekor minden teszt előállít egy értéket, melyeket aztán levelenként összegez és az előre beállított határérték alapján dönti el mi a spam és mi nem az (ham, azaz nem spam).

A Clam Antivirus program egy GPL licenc alatt fejlesztett program, mely széles körben elterjedt mind vállalati, mind otthoni felhasználók között. Vírusdefiníciós állományait gyakran napi több alkalommal is frissítik.

Az úgynevezett szürke lista (graylist) szerveroldali spamszűrő módszer, mely úgy működik, hogy a Postfix a kapott levelet először átmeneti hibával elutasítja, majd mikor a küldő bizonyos idő elteltével újraküldi azt, akkor már elfogadja. Előnye, hogy nincs téves pozitív azonosítás, hátránya, hogy sok levelező szerver nincs bekonfigurálva automatikus újraküldésre, továbbá nagy forgalmú helyeken nem alkalmazható, mert „eldugulhat” tőle a levelezés.

Többféle megoldás létezik erre a módszerre, én a Postgrey-t választottam, mert nincs szükség hozzá külső alkalmazásra (pl. adatbázis-kezelőre), egyszerű, könnyen telepíthető és konfigurálható. PERL nyelven íródott, Berkeley DB adatbázisban tárolja feladó/címzett/kliens adatokat. Telepítése:

```
apt-get install clamav-base libclamav5 clamav-daemon clamav-freshclam
amavisd-new spamassassin spamc postgrey
```

2.9 Webmail

Személyes kedvenc témámhoz értem. Sokak számára furcsának tűnhet, ha azt mondom, nem használok jó ideje, azaz idestova 5 éve levelezéshez kliens programot, szinte csak webmailt. Azt, hogy miért, könnyű lehet megérteni akkor, ha azt mondom, hogy nem szeretek géphez kötött lenni, szeretem elérni leveleimet, ha a saját gépem nincs is velem, mindezt biztonságos módon. Hiába rendelkezek sok mobil eszközzel, melyek internet képesek, számomra fontos szempont, hogy minden esetben ugyanazt a felületet kapjam amikor levelet akarok küldeni, fogadni vagy keresni a meglévőek tartalmában.

Több felület áll rendelkezésünkre, úgy mint a Squirrelmail, Roundcube, Horde-IMP csak hogy a legismertebbeket említsem. Személyes kedvencem a Roundcube, mely egy még stabil kiadás előtt álló eszköz, de gyorsaságával, könnyű kezelhetőségével (AJAX használata a felületen a levelek mozgatásához) kitűnik a sorból. Azonban sajnos a Debian Lenny-ben nem elérhető csomagból, kézzel kell azt telepítenünk, így használatától jelen esetben eltekinthetünk. A Horde keretrendszer az IMP webmaillel egy robusztus, sok modullal és kiegészítővel rendelkező felületet kínál, mint a címjegyzék kezelő modul, a Turba, az Ingo levélszűrést kezelő modul és a Mimp, mely mobileszközökön való megjelenést tesz lehetővé felhasználó-barát módon. Azonban inkább mutat az irodai munkát összefogó, koordináló program irányába, mint egyszerű levelezést lehetővé tevő program felé, konfigurálása, beállítása sokszor nehézkessé válhat.

Ezért esett választásom a Squirrelmail-re, mely a többihez hasonló módon rendelkezik magyar fordítással, rengeteg szolgáltatása van, gyors, könnyen kezelhető, képes IMAP-on keresztüli kommunikációra. A szolgáltatás listát bővítmények formájában adhatjuk hozzá a programhoz. Minden olyan kiszolgálón fut, ahol van PHP futtatási modul.[14]

```
apt-get install squirrelmail squirrelmail-locales php-pear php5-cli
```

2.10 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.[15] Telepítése:

```
apt-get install phpmyadmin
```

2.11 Tűzfal

Nélkülözhetetlen feltétel az internetes környezetben a tűzfal használata a biztonságos kommunikáció eléréséhez. A tűzfal olyan eszköz, mely a kommunikációs végpontok közötti forgalmat bizonyos előre beállított szabályok szerint engedélyezi vagy tiltja. A hálózati kommunikáció legkisebb egysége az adatcsomag, melynek fejrésze tartalmazza a kézbesítéshez és kezeléséhez szükséges információkat, úgy mint forrás és cél IP cím, portok, protokollok azonosítói. Az adatmező része pedig az átvinni szánt felhasználói adatokat tartalmazza. Az összetartozó adatcsomagok kommunikációs csatornát alakítanak ki. Három -féle tűzfaltípust különböztetünk meg, az első ilyen az úgynevezett csomagszűrő, mely adatcsomag szinten foglalkozik a rajta keresztül áramló információval, nem vizsgálva azt, hogy milyen kommunikáció része. Tehát a csomagszűrő minden egyes csomagról külön döntést hoz. Ennek megfelelően a szűrési szabályok is csak a csomag fejrészében található információkra vonatkoznak. Ez alapján megadható szabálynak, hogy mely címekről vagy mely címekre engedélyezzük a továbbítást, milyen portokat és protokollokat engedélyezünk. Ezen tűzfalak csak nagyon egyszerű szűrést tudnak megvalósítani, hatékony védelemre nem alkalmasak.

Az állapotfigyelő tűzfalak fejlettebb megoldást nyújtanak. Szintén a fejrészben található információk alapján hoznak döntést, de nyilvántartják, hogy az adott csomag mely

kommunikációs kapcsolat része, s amikor a csomag továbbításáról döntenek, képesek figyelembe venni a csomag kapcsolaton belüli szerepét. Tehát szűrési szabályainak megadásakor hivatkozhatunk a csomag kommunikációs csatornán belüli szerepére is a fejrész információkon felül.

A legmagasabb szintű kontrollt a hálózati réteg felett az úgynevezett alkalmazási réteg szintű tűzfalak valósítják meg. Ezek már a csomagok adatmezőiben is vizsgálatot végeznek, azaz az ott szállított információt összefüggő adatfolyamnak tekintve alkalmazás-szintű szűrést is végezhetnek. Tehát képesek megszűrni a böngészés közben keletkező adatfolyamból a rosszindulatú tartalmakat, levelek káros részeit megszűrni, fájlok átvitelét különböző webes applikációkban.

Tűzfalnak a Linux kerneljében található netfilter tűzfal funkciót választom. Ez működhet egyszerű csomagszűrőként vagy állapotfigyelő szűrőként is. Az utóbbi az alapértelmezett, így én is ezt fogom használni. Konfigurálása az iptables parancson keresztül fog történni. Ahhoz, hogy a beépített tűzfal működését a saját igényeinkhez igazítsuk, a csomagokra szabályokat kell meghatároznunk. A szabályok megadása az iptables parancssori programmal történhet. Ennek könnyebb kezelhetősége végett választottam hozzá egy adminisztrációs felületet, a Shorewall-t.

A Shorewall egy iptables konfiguráló program, egy megbízható iptables beállítást ír, melyet alkalmazhatunk rendszerünkön.[16] Telepítése:

```
apt-get install shorewall shorewall-perl
```

3 Konfigurálás

Elsőként a rendszeridő beállításáról gondoskodunk:

```
apt-get install ntpdate
```

Ezután gépünk automatikusan szinkronizálni fogja az időt publikus szerverekről, melyek a „`/etc/default/ntpdate`” fájlban vannak definiálva.

3.1 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',
```

```
PRIMARY 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.

```
vim /etc/mysql/my.cnf
log = /var/log/mysql/mysql.log
```

Érvénybe léptetéséhez újra kell indítani a szolgáltatást:

```
/etc/init.d/mysql restart
```

3.2 Postfix konfigurálása

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.[17] 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. [3]

A szerver FQDN nevét el kell helyezni az alábbi fájlban.

```
vim /etc/mailname
mail.pelda.hu
```

Nyissuk meg szerkesztésre a postfix main.cf nevű fájlját.

```
vim /etc/postfix/main.cf
```

Adjuk meg az internet hostnevet a gépünknek.

```
myhostname= mail.pelda.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 (Debian/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:

•korlátozási szakasz

•korlátozás

•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

HELO/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.

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

Felhasználói fiókok helyét tartalmazó adatbázisbeli tábla helyét leíró fájl megadása.

```
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 /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.

```
vim /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.

```
vim /etc/postfix/mysql_uid.cf
user=mail
password=qwer1234
dbname=maildb
table=users
select_field=uid
```

```
where_field=id
hosts=127.0.0.1
```

A csoportazonosító meghatározásához.

```
vim /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.

```
vim /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.

```
vim /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
```

3.3 Courier beállítása

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:

```
vim /etc/courier/authdaemonrc
authmodulelist="authmysql"
DEBUG_LOGIN=2
```

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

```
vim /etc/courier/authmysqlrc
MYSQL_USERNAME mail
MYSQL_PASSWORD qwer1234
MYSQL_DATABASE maildb
MYSQL_USER_TABLE users
MYSQL_CRYPT_PWFIELD crypt
# MYSQL_CLEAR_PWFIELD clear
```

3.4 Adatfeltöltés

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 ('pelda.hu');
INSERT INTO aliases (mail,destination) VALUES ('@pelda.hu',
'postmaster@pelda.hu'), ('postmaster pelda.hu','postmaster@pelda.hu'),
('abuse@pelda.hu','postmaster@pelda.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”[18] 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@pelda.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/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.

3.5 Tartalom szűrők beüzemelése

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

Tegyük megjegyzéssé a vírus és spam figyelését egyelőre.

```
vim /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).

```
vim /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.

```
vim /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.

```
vim /etc/default/spamassassin
ENABLED=1
vim /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.

```
vim 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;
```

3.6 Autentikáció

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.[19]

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

A Postfix-hez adjuk hozzá a SASL konfigurációt.

```
vim /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 meglévő 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.

```
vim /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.

```
vim /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.

```
/etc/init.d/saslauthd restart
/etc/init.d/postfix restart
```

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.

```
vim /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.

```
/etc/init.d/courier-authdaemon restart
/etc/init.d/courier-imap restart
```

3.7 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@pelda.hu
```

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

```
vim /etc/postfix/main.cf
#smtp_use_tls = no
smtp_tls_security_level = may
```

```
#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.

```
vim /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_destinati
on,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.

```
vim /etc/courier/imapd-ssl
TLS_CERTFILE=/etc/courier/imapd.pem
```

3.8 Webmail

Másoljuk be a Squirrelmail Apache-hoz készült konfigurációs fájlját az Apache konfigurációs könyvtárba, majd engedélyezzük és töltessük be vele azt.

```
cp /etc/squirrelmail/apache.conf /etc/apache2/sites-available/squirrelmail
a2ensite squirrelmail
/etc/init.d/apache2 reload
```

Nézzük át és állítsuk be a felületet következő parancs futtatásával, mely egy menüs konfigurációs felületet ad.

```
squirrelmail-configure
```

Válasszuk ki a 2-es menüt, azon belül pedig az A pontot. A 8-as menünél (Server software) írjuk be, hogy „courier”.

Majd ugyanitt a 7-es menüpontot állítsuk „TRUE”-ra, azaz engedélyezzük a TLS-t, majd a 7-es menünél adjuk meg az IMAP portot, a „993”-at.

Állítsuk be az adatbázis kapcsolathoz a paramétereket a 9-es menüpontban. Az 1-es és 3-as ponthoz adjuk meg a következőt:

```
mysql://mail:qwer1234@127.0.0.1/maildb
```

Lépjünk ki a beállítások elmentése után, majd ezen beállításoknak megfelelően hozzuk létre az adatbázis táblákat a címlistához.

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

Először a felhasználó címlistáját tárolót.

```
CREATE TABLE `address` (
 `owner` varchar(128) NOT NULL default '',
 `nickname` varchar(16) NOT NULL default '',
 `firstname` varchar(128) NOT NULL default '',
 `lastname` varchar(128) NOT NULL default '',
 `email` varchar(128) NOT NULL default '',
```

```
 `label` varchar(255) default NULL,
 PRIMARY KEY (`owner`,`nickname`),
 KEY `firstname` (`firstname`,`lastname`) ) ;
```

Majd a preferencia táblát.

```
CREATE TABLE `userprefs` (
 `user` varchar(128) NOT NULL default '',
 `prefkey` varchar(50) NOT NULL default '',
 `prefval` varchar(255) default NULL,
 `modified` timestamp(14) NOT NULL,
 PRIMARY KEY (`user`,`prefkey`) ) ;
```

Végezetül pedig a globális, mindenki által elérhető címeket tároló táblát.

```
 CREATE TABLE `global_abook` (
 `owner` varchar(128) NOT NULL default '',
 `nickname` varchar(16) NOT NULL default '',
 `firstname` varchar(128) NOT NULL default '',
 `lastname` varchar(128) NOT NULL default '',
 `email` varchar(128) NOT NULL default '',
 `label` varchar(255) default NULL,
 PRIMARY KEY (`owner`,`nickname`),
 KEY `firstname` (`firstname`,`lastname`) );
```

3.9 Tűzfal

A tűzfal beállítása az első lépés, fontos, hogy már a telepítés folyamatában megfelelő módon le legyen zárva a külvilágtól és csak a legszükségesebbek maradjanak nyitva. Ennek fényében bezárok minden portot, kivéve az SSH kapcsolódáshoz szükségeset. A későbbiekben szükséges lesz a teszteléshez majd az üzemeltetéshez megnyitni további portokat, ezeket a telepítés megfelelő fázisaiban fogom megtenni.

Alapértelmezettként a Shorewall nem rendelkezik beállításokkal, nekünk kell az igényekhez megfelelően testre szabni. A „`/usr/share/doc/shorewall-common/default-config`” könyvtár tartalmazza az alapértelmezett konfigurációs fájlokat, melyeket a „`/etc/shorewall`”-ba másolva és szerkesztve elkészíthetjük az igényeinknek megfelelő

konfigurációt. Segítségképpen a „/usr/share/doc/shorewall-common/examples”-ben találhatunk minta konfigurációkat, vagy a man page-ben nézhetünk utána a fájlnak (például.: „**man shorewall-hosts**”).

Elsőként megadjuk, mely hálózati interfészen keresztül csatlakozunk az internetre. Az „interfaces” állományban lehet meghatározni, hogy melyik interface melyik zónához tartozik. A shorewall alapértelmezetten három zónát kezel: „fw”, „loc”, „net”

fw zóna – Az „fw” zóna az maga az átjáró / tűzfal gép localhostját jelöli.

loc zóna – A „loc″ zóna a lokális hálózat meghatározó neve.

net zóna – A „net″ zóna a magát az internetet vagy külső hálózatot jelöli.

A fájl formája: „ZONE INTERFACE BROADCAST OPTIONS”

```
cp /usr/share/doc/shorewall-common/default-config/interfaces
/etc/shorewall/
vim /etc/shorewall/interfaces
```

Adjuk meg a következő beállításokat benne.

```
#ZONE    INTERFACE      BROADCAST       OPTIONS
#
net       eth0            detect \      tcpflags,tcpflags,nosmurfs
```

Majd a hálózati zónákat állítjuk be.

```
cp /usr/share/doc/shorewall-common/default-config/zones /etc/shorewall/
vim /etc/shorewall/zones
```

Adjuk hozzá a következőt:

```
#ZONE  TYPE            OPTIONS         IN                      OUT
#                                      OPTIONS                 OPTIONS
fw firewall
net ipv4
```

Figyeljünk, hogy szerepeljen a tűzfalra vonatkozó sor:

```
fw firewall
```

Definiáljuk az alapértelmezett szabályokat a „policy” fájlban. Ez a Shorewall egyik legfontosabb állománya, beállításai az összes beállításra érvényes és kihatnak. Itt lehet meghatározni a globális tulajdonságokat, azaz itt lehet megadni, merre engedünk és merre nem engedünk forgalmat és ez milyen formában legyen naplózva. A szabályokat sorról sorra

fentről lefele veszi és az első illeszkedőt alkalmazza. A fájl a következő mezőket tartalmazza adott sorrendben: „CLIENT SERVER POLICY LOGLEVEL” .

Client – Az „interface” állományban megadott egyik zóna, ez lesz a „honnan” azaz a forrás zóna. Lehet: „fw”, „loc”, „net”, „dmz”

Server – Az „interface” állományban megadott egyik zóna ez lesz a „hova” azaz a cél zóna. Lehet: „fw”, „loc”, „net”

Policy – Itt határozhatjuk meg a „honnan hová” tulajdonságait. Lehet: „ACCEPT”, „DROP”, „ REJECT”

Log – Itt határozhatjuk meg a naplózást. Ami lehet „info” vagy „kernel”. Ajánlott az „info”, és csak a „DROP REJECT”-re különben rengeteg felesleges naplóbejegyzés lesz engedélyezett csomagra, ami rengeteg helyet is elfoglalhat.

A „Client” és „Server” kapcsolók jelentései az alábbiak.

fw – Helyi gép az átjáró vagy a tűzfal meghatározása.

loc – Helyi vagy routolt hálózat.

net – A külső hálózat, vagy internet.

dmz – Az a zóna, amelyikkel idegen hálózat gépei kapcsolatot létesíthetnek.

A „Policy” kapcsolóinak pedig az alábbiak.

Accept – Az összes csomag elfogadása.

Drop – Az összes csomag eldobása, megtagadása.

Reject – Az összes csomag eldobása, megtagadása, annyi különbséggel, hogy a csomag küldője „port unreachable” hibaüzenetet kap. Gondolni kell arra is, hogy bizonyos esetekben ez az ICMP hibaüzenet nem kerül elküldésre (lásd: RFC 1122):

•A kiszűrt csomag egy ICMP hibaüzenet, vagy ismeretlen ICMP típusú volt.

•A kiszűrt csomag egy fejléc nélküli töredék volt.

•Túl sok ICMP hibaüzenetet küldtünk erre a címre mostanában.

```
cp /usr/share/doc/shorewall-common/default-config/policy /etc/shorewall/
vim /etc/shorewall/policy
$FW net ACCEPT
net $FW DROP info
net all DROP info
#Elutasításra kerül minden más:
all all REJECT info
```

Leállás esetére pedig:

```
cp /usr/share/doc/shorewall-common/default-config/routestopped
/etc/shorewall/
vim /etc/shorewall/routestopped
eth0 0.0.0.0 routeback
```

A tűzfal szabályokra pedig: előregyártott makró szabályok találhatók a Shorewall-ra a „`/usr/share/shorewall`”-ban. A „`rules`” állománynak a feladata a tulajdonképpeni iptables szabályok, körök meghatározása.

Beállítása attól függ, hogy a „`policy`” állományban milyen tulajdonságokat adtunk meg a „CLIENT SERVER” meghatározásoknak. Itt lehet valójában portokat nyitni zárni, átirányítani. Minden névre a „policy” állomány szerint kell hivatkozni. Néhány Shorewall változatban az fw zónát, $FW-nek kell megadni (ahogyan azt Debian Linux alatt is).

```
cp /usr/share/doc/shorewall-common/default-config/rules /etc/shorewall/
vim /etc/shorewall/rules
SSH/ACCEPT net $FW
```

4 Tesztelés

Leggyakoribb előforduló hiba az elgépelés. Ha kész kódrészletet emelünk át valahonnan, előfordulhat, hogy a sortöréseket nem tartja meg vagy olyan karaktereket emel át, amit nem tud értelmezni az feldolgozó, mint például a Windows rendszerből származó sortörés karakter, amennyiben ezen operációs rendszeren keresztül konfiguráljuk rendszerünket. Nem túl gyakori hiba, de megeshet, hogy az adott eszköz újabb verziója már más szintaktikát követ egy korábbihoz képest, ezért minden esetben érdemes átnézni az aktuális dokumentációját.

Telepítés közbeni folyamatos tesztelésnek – azaz hogy működik-e a küldés és fogadás – előnye, hogy azonnal detektálható, hol van a hiba. Ezzel a probléma izolációs eljárással sok idő elmegy, talán több is, mint magával a telepítés folyamatával, azonban nem szabad elsiklani felette, hiszen ezzel a későbbiekre vetítve sok időt és energiát spórolhatunk meg magunknak. Egyúttal jobban képbe is kerülünk magával a rendszerrel és annak működésével, ami nem hátrány a későbbi üzemeltetés szempontjából. A hibákat ugyanis nem lehet előre

determinálni, nem lehet őket sablonszerűen kezelni. Legtöbbször azonban az első nehézséget az okozza, hogy megtaláljuk, pontosan mi okozza a hibát. Ebben nyújt nekünk segítséget az egyes eszközök naplózási funkciója, melyeket sokszor a teljesítmény növelése végett kikapcsolunk, vagy kisebb szintre állítunk, majd hiba esetén újra használatba vesszük őket.

A hiba keresését mindig a naplófájlok átnézésével kezdjük valamilyen szűrés beiktatásával. A leggyakrabban tallózott naplófájlok az alábbiak:

„`/var/log/system.log`”

„`/var/log/mail.log`”

„`/var/log/mysql.log`”

„`/var/log/apache2/access.log`”

Szükség lehet még a szolgáltatások ki- és bekapcsolására:

```
/etc/init.d/apache2 stop/start
/etc/init.d/courier-imap-ssl stop/start
/etc/init.d/courier-imap stop/start
/etc/init.d/courier-authdaemon stop/start
/etc/init.d/courier stop/start
/etc/init.d/postfix stop/start
/etc/init.d/postgrey stop/start
/etc/init.d/amavisd stop/start
/etc/init.d/spamassassin stopv
/etc/init.d/clamav-daemon stop/start
/etc/init.d/clamav-freshclam stop/start
/etc/init.d/mysql stop/start
```

Bekapcsolhatjuk a tűzfal naplózását is. Adjuk hozzá az összes „`REJECT`”, „`BOUNCE`” és „`DROP`” szabályhoz az „`INFO`” kapcsolót a policy fájlban az alábbi módon:

```
vim /etc/shorewall/policy
net all DROP info
```

A Courier naplózásához szerkesszük az „`authdaemonrc`” fájlt:

```
vim /etc/courier/authdaemonrc
DEBUG_LOGIN=2
```

A Courier-IMAP naplózásának bekapcsolása:

```
vim /etc/courier/imap
DEBUGLEVEL=2
```

Van lehetőség az Amavis vizsgálatára is:

```
vim /etc/amavis/conf.d/50-user
$log_level = 2;
```

Az elhagyhatatlan eszközünk teszteléskor a „**telnet**” parancs. Segítségével szimulálható a levélküldés lokálisan, melynek eredményeit a naplófájlok elemzésére használhatjuk. Ezáltal kiszűrhető a tűzfal és hálózati probléma, vizsgálható, hol akad el a levélkézbesítés.

```
telnet localhost 25
```

Az EHLO kézfogás elküldése a kapcsolat kiépítése céljából. Adjunk meg itt egy valós tartománynevet, ahonnan a levél küldője érkezhet.

```
EHLO mail.example.com
```

A szerver ad válaszként pár sort, mint például:

```
>250-mail.example.com
>250-PIPELINING
```

Adjuk meg, hogy kitől érkezik a levél.

```
MAIL FROM: <your@example.com>
```

Majd adjunk meg egy helyi felhasználót.

```
RCPT TO: <local_user@example.org>
```

Következhet a levél adatrészének megadása a „data”-val. Itt lehet megadni továbbá a levél tárgyát is a „Subject: tárgy” formában. Lezáráshoz tegyünk csak egy pontot az utolsó sorba.

```
data
Subject: teszt levél
Ez itt most egy próbaként küldött levél.
.
quit
```

Közben érdemes a hibák megtalálásának érdekében a „/var/log/mail.log” és „/var/log/mysql.log” fájlt figyelni az esetleg felmerülő hibák észrevételéhez.

```
„tail -f /var/log/mail.log”
„tail -f /var/log/mysql.log”
```

Általánosságban előforduló hibák és megoldásaik:

•Nem válaszol a szerver a 25-ös porton

•Nincs szolgáltatás a port mögött:

◦ „netstat -ptn” utasítással nézzük meg, fut-e a Postfix

◦ ellenőrizzük a tűzfal beállításait és naplófájljait

•Feladó tartománynevét nem fogadja el

•Valós tartományt használjunk tesztelésre is

◦ figyeljünk az EHLO és RCPT TO: helyes megadására a „**telnet**” használatakor

◦ DNS névfeloldás nem működik, próbáljunk pingelni egy szervert a helyi gépről

•A Postfix fogadja a levelet, de a MySQL naplófájlokban nem jelenik meg

•A MySQL kapcsolattal van a baj

◦ MySQL socket meglétének ellenőrzése

◦ Elgépelés a Postfix konfigurációs fájlok valamelyikében

◦ Postfix konfigurációs fájlok jogosultsági problémája

◦ MySQL konfigurációs fájlokban a „`host`” mező cseréje „`localhost`” és „`127.0.0.1`” között

•Vírusszűrő nem megfelelő működése

◦ a konfigurációs fájlokban szintaktikai hiba van. Ellenőrizhető a következő paranccsal: „**spamassassin --lint**”

◦ parancssorból tesztelhető a futása:

„**spamassassin -D < /usr/share/spamassassin/doc/spamassassin-3.0.3/sample-spam.txt**”

A szolgáltatások újraindításával is hasznos információkhoz juthatunk a naplófájlok figyelésével. Előfordulhat, hogy valamelyik nem megfelelően indul el, mely okozhat nem megfelelő működést.

Szükség lehet megvizsgálni, hogy a rendszerből tudunk-e levelet küldeni mások irányába.

```
telnet localhost 25
```

Az EHLO kézfogás elküldése a kapcsolat kiépítése céljából. Adjunk meg itt a helyi tartománynevet, ahonnan a levelet fogjuk küldeni.

```
EHLO example.org
```

Adjuk meg a helyi felhasználót akinek a nevében küldjük ki a levelet.

```
MAIL FROM: <local_user@example.org>
```

Majd akinek szeretnénk a levelet küldeni.

```
RCPT TO: <anyone@example.com>
```

Következhet az adatrész.

```
data
Subject: teszt levél
Ez is egy próba levél.
.
quit
```

Amennyiben nem kapunk hibát, a levélküldés sikeres, különben következhet a naplózás átnézése. Sikeres küldés esetén valami hasonlónak kell szerepelni a naplófájlokban:

```
Dec 17 10:25:45 servername postfix/smtp[12345]: 12345678:
to=<you@example.com>, relay=127.0.0.1[127.0.0.1]:10024, delay=15,
delays=15/0.01/0.02/0.11, dsn=2.0.0, status=sent (250 2.0.0 Ok, id=12345-
09, from MTA([127.0.0.1]:10025): 250 2.0.0 Ok: queued as 1234567)
```

Ha helyi gépről parancssorból sikeresen tudunk küldeni és fogadni leveleket, érdemes a webes felületről vagy egy kliens levelezőből tesztelni a levelezést.

A fenti konfiguráció lehetőséget biztosít számunkra arra az esetre, ha szeretnénk előre elkészített sablon szerver telepítéseket használni. Ebben az esetben telepítjük az alaprendszert majd ezt követően a fenti alap levelezőt és elkészítjük a képfájlt a szerverről. Erre az esetre következzen pár pont és azok kifejtése, miket kell átnézni és személyre szabni a konfigurációban.

•Állítsunk le minden szolgáltatást

•Korlátozzunk a tűzfalon a forgalmat

•Változtassunk jelszavakat

•Nézzük át a konfigurációs fájlokat

•Állítsuk be a gépnevet

•A tanúsítványokat készítsük el

•Indítsuk el és teszteljük a szolgáltatásokat

•Töltsük fel adatokkal

•Engedélyezzük a tűzfalon újra a forgalmat

•Teszteljünk

Állítsuk le minden szolgáltatást

```
/etc/init.d/postfix stop
/etc/init.d/courier-imap-ssl stop
/etc/init.d/courier-imap stop
```

```
/etc/init.d/courier-authdaemon stop
/etc/init.d/mysql stop
/etc/init.d/amavisd stop
/etc/init.d/spamassassin stop
/etc/init.d/clamav stop
```

Korlátozzuk a tűzfalon a forgalmat.

Az ssh forgalmat engedélyezzük csak mindaddig, amíg minden beállítással nem végzünk. Ehhez szerkesszük a rules fájlt úgy, hogy csak az ssh részt nem tesszük megjegyzésbe.

```
vim /etc/shorewall/rules
```

Változtassuk jelszavakat.

Valószínűleg az új rendszernél szeretnénk más jelszavakat használni, mint amiket a sablon elkészítésekkor megadtunk.

Változtassuk meg azon felhasználók jelszavait a rendszerben akiket esetlegesen létrehoztunk és a jelen rendszerben is szükségesek, a többit távolítsuk el.

Ezen felhasználók „`ssh`” hozzáférését is ellenőrizzük le a „`/etc/ssh/sshd_config`” fájlban.

Az adatbázis felhasználók jelszavainak módosítását kezdjük a „**root**” felhasználóéval.

```
mysqladmin -u root password új_jelszó -p
```

A módosításához meg kell adnunk a telepítéskor megadottat.

A „**mail**” adatbázis felhasználó jelszavát is módosítsuk.

```
mysqladmin -u mail password new_password -p
```

Abban az esetben ha nem tudnánk mit adtunk meg korábban, akkor lépjünk be a „**root**” felhasználóval és módosítsuk parancssorból.

```
mysql -u root -p
update mysql.user set password=password('új_jelszó') where user='mail';
flush privileges;
```

Ezen változtatások után minden Postfix-ben megadott jelszavat is módosítani kell az összes releváns fájlban.

```
vim /etc/postfix/mysql*
```

A következő változót kell keresnünk a fájlokban és megadni neki az új értéket:

```
password=új_jelszó
```

A Courier adatbázis fájlját is frissítsük a megfelelő érték megadásával.

```
vim /etc/courier/authmysqlrc
MYSQL_PASSWORD új_jelszó
```

Nézzük át a konfigurációs fájlokat

Érdemes minden fentebb említett konfigurációs állományt átnézni annak érdekében, hogy minden beállítás a jelenlegi igényekhez legyen igazítva.

Állítsuk be a gépnevet

Adjuk meg ez új gép által használt gépnevet, „gépnév.új_domainnév.tld” formában.

```
vim /etc/hostname
```

Majd ezt adjuk meg, mint levelező nevet is.

```
cp /etc/hostname /etc/mailname
```

A gép által kezelt lokális „/etc/hosts” állományba is vegyük fel vagy módosítsuk a meglévőt.

```
vim /etc/hosts
```

Tartalmazza az alábbi sort:

```
127.0.0.1 gépnév.új_domainnév.tld gépnév
```

Ezt az újonnan beállított majd használni kívánt nevet meg kell adni a Postfix-nek is.

```
vim /etc/postfix/main.cf
```

Az alábbi változó értékeként, mely nem azonos a gép által használt névvel természetesen.

```
myorigin=új_domainnév.tld
```

A tanúsítványokat készítsük el

Ha használni fogunk az új rendszeren tanúsítványokat, akkor ha készítettünk a sablon elkészítéséhez, cseréljük azokat, egyébként pedig készítsünk újakat.

Indítsuk el és teszteljük a szolgáltatásokat

```
/etc/init.d/postfix start
/etc/init.d/courier-imap-ssl start
/etc/init.d/courier-imap start
/etc/init.d/courier-authdaemon start
/etc/init.d/mysql start
/etc/init.d/amavisd start
/etc/init.d/spamassassin start
/etc/init.d/clamav start
```

A tesztelésnél leírtak alapján nézzük végig, hogy minden szolgáltatás fut és ellátja feladatát.

Töltsük fel adatokkal

A leírtak szerint adjuk hozzá az összes szükséges adatot az adatbázishoz.

Engedélyezzük a tűzfalon újra a forgalmat

Nyissuk meg a világ felé az összes szükséges portot ami a szolgáltatás működéséhez szükséges és elégséges.

Teszteljünk

Elengedhetetlen, hogy mind parancssorból és külső kliensről is egyaránt lefuttassuk a testeket, hogy meggyőződhessünk az általunk kívánt működésről.

5 Összefoglalás, továbbfejlesztési lehetőségek

Fenti instrukciókat követve egy egyszerű, könnyen karbantartható, biztonságos levelező rendszert kapunk, melyet a tesztelések végeztével azonnali használatba vehetünk. Biztonsága adódik abból, hogy csak a legszükségesebb csomagokat helyeztük el benne, a szolgáltatásokat pedig tűzfal mögé rejtettük, a nyitva marad szolgáltatásokat pedig a hitelesítéssel tettük biztonságossá.

Fontos kiemelnem, hogy a mindennapi üzemeltetési tapasztalat az mutatja, hogy nap mint nap követnünk kell az újonnan megjelenő, kéretlen levélküldők ellenében megjelenő technikákat. Ugyanis a legtöbb erőforrást az ellenük alkalmazott módszerek viszik el a rendszerből, úgymint például a greylisting, melyet igen jó eredményekkel lehet alkalmazni kisebb levelezőknél, míg nagy forgalmat bonyolítóknál nem, mert a nagyszámú levelek átnézésével a rendszer nagy késleltetési idővel tud csak dolgozni, ami a mai rohanó világban nem megengedhető.

Így ez a könnyen testre szabható, nagy egyedi szabadságot nyújtó rendszer erős alapja lehet bárki által bárhol használható levelező rendszernek.

Összetettsége miatt számos lehetőség kínálkozik a továbbfejlesztésre.

- A teljes rendszert egy virtuális környezetbe telepítjük, mint például a XEN vagy KVM Linux alapú virtualizációs technológiák. Így egy még flexibilisebb, könnyebben hangolható és klónozható rendszert kapunk.
- Nagy rendelkezésre állású tárolókat csatolhatunk a rendszerhez.
- Másodlagos SMTP szerver hozzáadása a rendszerhez (backup MX).

6 Köszönetnyilvánítás

Köszönöm témavezető tanáromnak, Ecsedi Kornélnak, hogy bölcs meglátásaival s szakmai kérdésekben nyújtott válaszaival segítette munkámat, továbbá munkáltatómnak, hogy a teszt rendszerek létrehozásához szükséges infrastruktúrát biztosította számomra.

7 Rövidítések jegyzéke

•**HTTP** - HyperText Transfer Protocol; információátviteli, kérés-válasz alapú portokoll kliensek és szerverek között

•**HTTPS –** URI séma, jelzi, hogy a HTTP protokollt kell használni a szerver 443-as TCP portján a HTTP és TCP szintek közé titkosító/hitelesítő SSL vagy TLS réteg beiktatásával.

•**DNS** – A Domain Name Service (DNS) az egyik legfontosabb szolgáltatás az Interneten. Fő feladata a webcímek „lefordítása”, „feloldása” a hozzájuk tartozó IP-címre.

•**SMTP –** Kommunikációs protokoll az e-mailek Interneten történő továbbítására. Először a 821-es RFC-ben definiálták, majd kiegészítésre került az RFC 5321-essel.

•**FQDN –** A Fully Qualified Domain Name = Hibátlan Teljes Domain-Nevet (FQDN) szokták *abszolút/teljes domain név*nek is hívni, ami egy olyan tartomány név, ami teljesen pontosan meghatározza a helyet a Domain Name System = Tartomány Név Rendszer (DNS) fa hierarchiájában.

•**POP3** – A Post Office Protocol version 3 (POP3) egy alkalmazás szintű protokoll, melynek segítségével az e-mail kliensek egy meglévő TCP/IP kapcsolaton keresztül letölthetik az elektronikus leveleket a kiszolgálóról. Napjainkban ez a legelterjedtebb protokoll az elektronikus levelek lekéréséhez. RFC 1939.

•**IMAP** – Az IMAP (Internet Message Access Protocol) egy alkalmazás rétegbeli protokoll, amely segítségével a leveleinkhez férhetünk hozzá. RFC 3501.

•**SSL** – Az SSL (Secure Socket Layer) egy protokoll réteg, amely a szállítási rétegbeli protokoll (pl. TCP/IP) és valamely alkalmazási rétegbeli protokoll (pl. HTTP) között helyezkedik el, az OSI terminológia szerinti viszony- és megjelenítési réteg feladatait látva el.

•**TLS** – A protokoll elsődleges célja a titkosság és az adatintegritás biztosítása a két egymással kommunikáló alkalmazás között. A protokoll két rétegből áll: a „TLS Record Protocol”-ból és a „TLS Handshake Protocol”ból. Az alacsonyabb szinten található a TLS Record Protocol, amely valamelyik megbízható átviteli protokollra épül (például a TCP-re). RFC 2246.

•**CLI –** Mozaikszó, az angol Command Line Interface (parancssoros felhasználói felület) rövidítése.

•**TLD** – A legfelső szintű tartomány (angolul top-level domain, TLD) a tartománynevek (domain nevek, doménnevek) végződésére, az utolsó pont utáni részre utal.

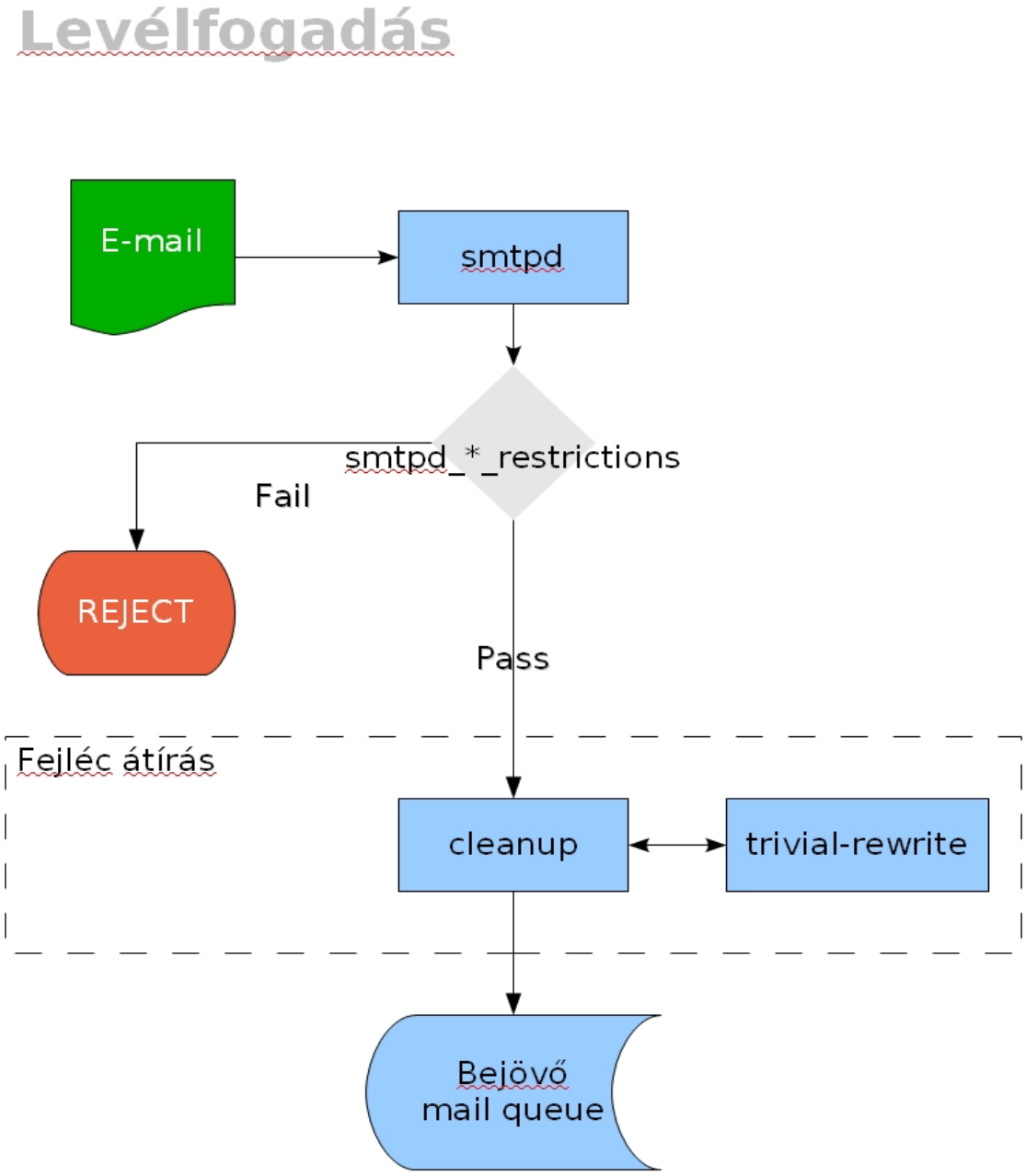
•**SASL** – A SASL a „Simple Authentication Layer” angol kifejezés rövidítése, egy azonosításra és adatbiztonságra szolgáló keretrendszer az internetes protokollban, és a 2222-es számú RFC taglalja részletesen.

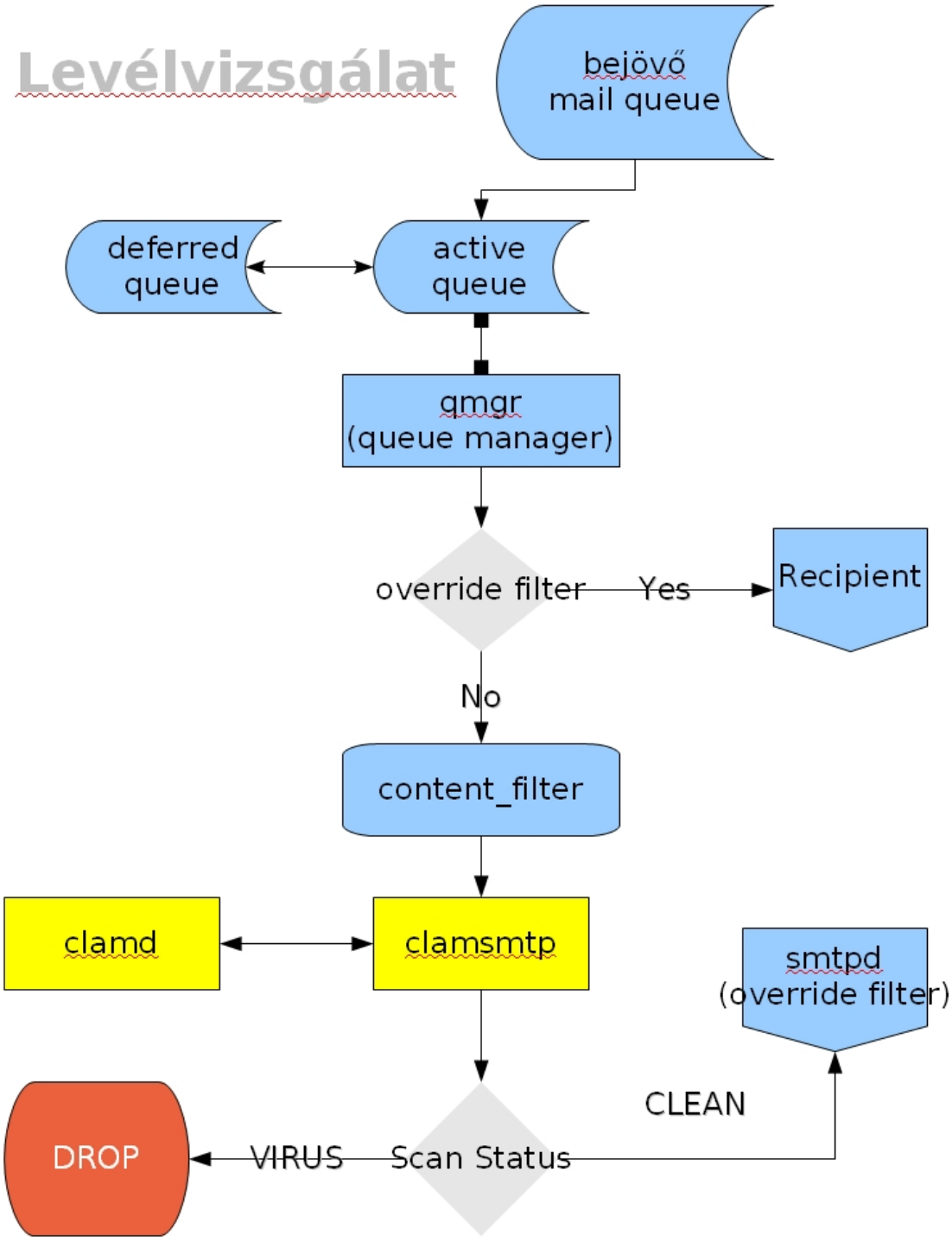
•**UCE –** Az UCE jelentése angolul: Unsolicited Commercial Email. Manapság inkább spam-nek ismerjük és sajnos már (majdnem) mindenhol találkozunk ilyen fajta reklámot híresztelő e-mailekkel.

•**PAM –** A PAM (Pluggable Authentication Modules; Csatlakoztatható Azonosítási Modulok) a felhasználói azonosítás csúcsa bármely modern Linux disztribúcióban.

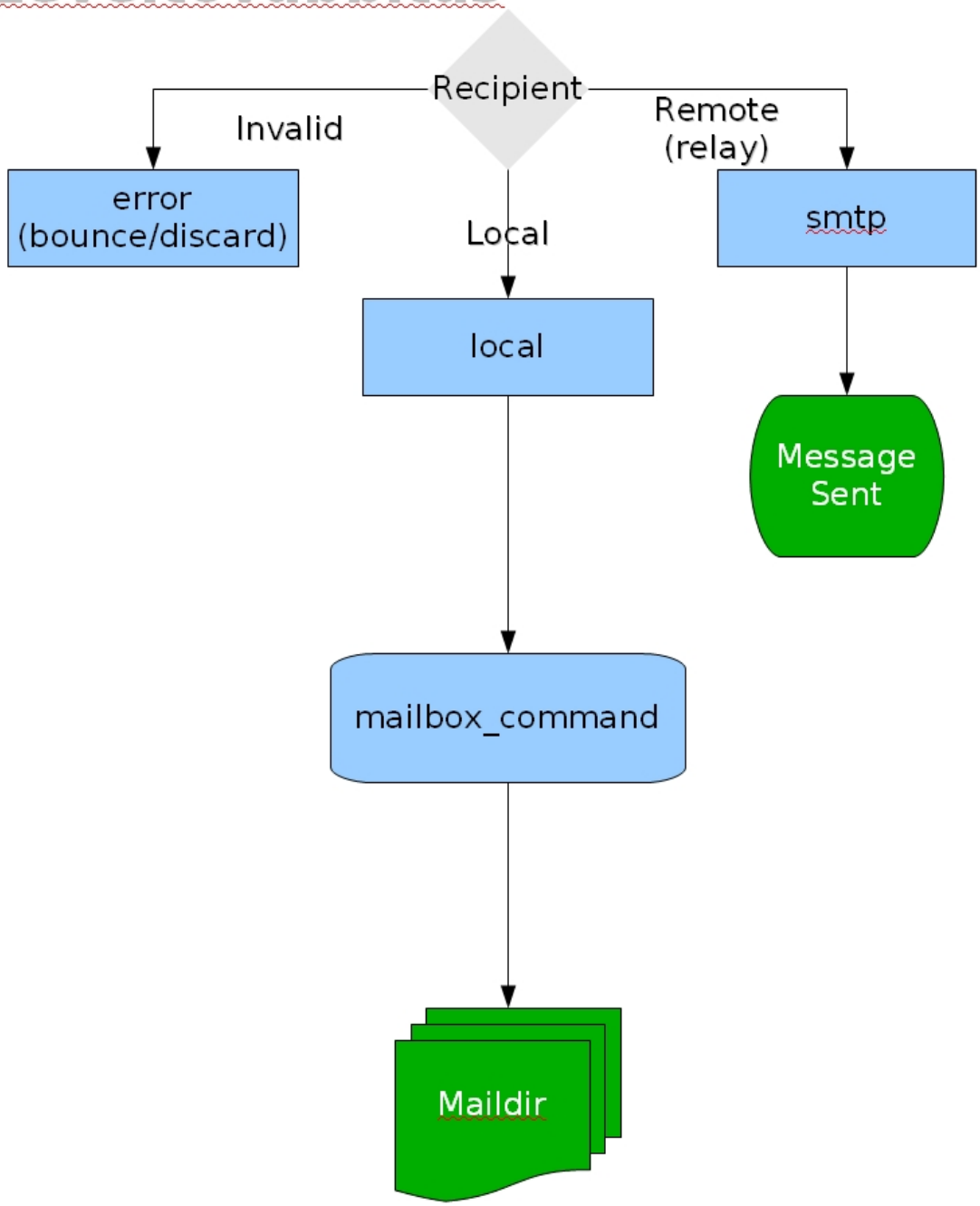
8 Mellékletek

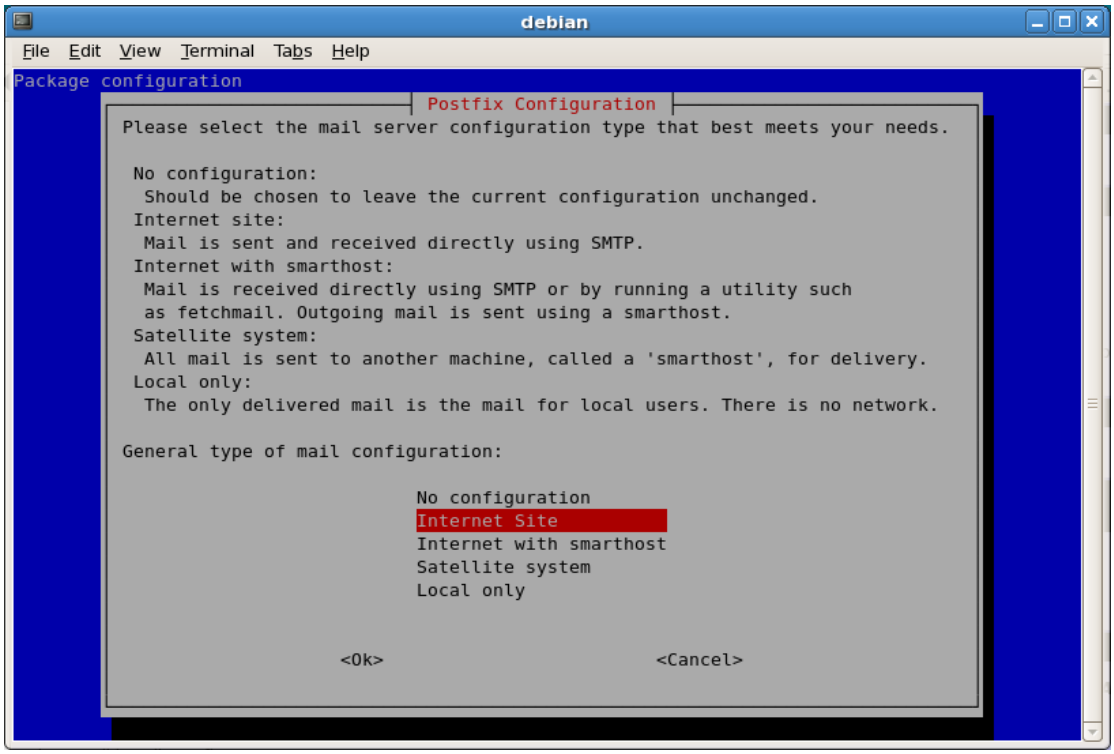
Működési diagramok



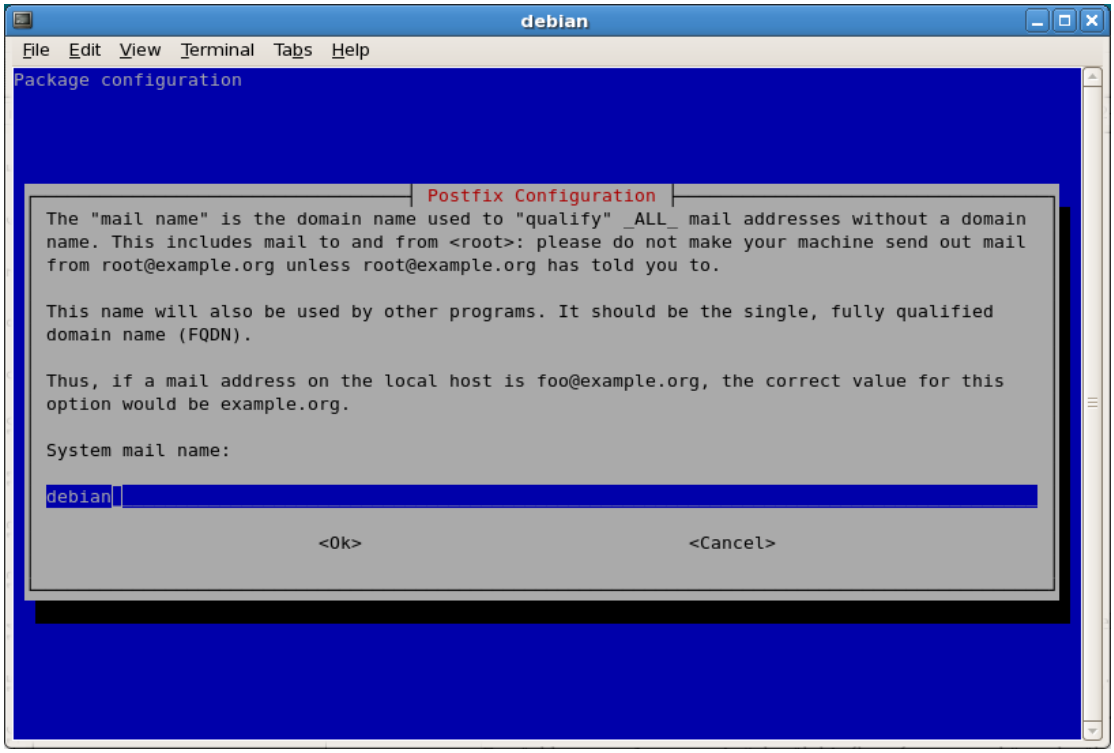

Levélvizsgálat
bejövő
mail queue
deferred
queue
active
queue
qmgr
(queue manager)
override filter
Yes
Recipient
No
content_filter
clamd
clamsmtp
smtpd
(override filter)
CLEAN
DROP
VIRUS
Scan Status

Levéltovábbítás

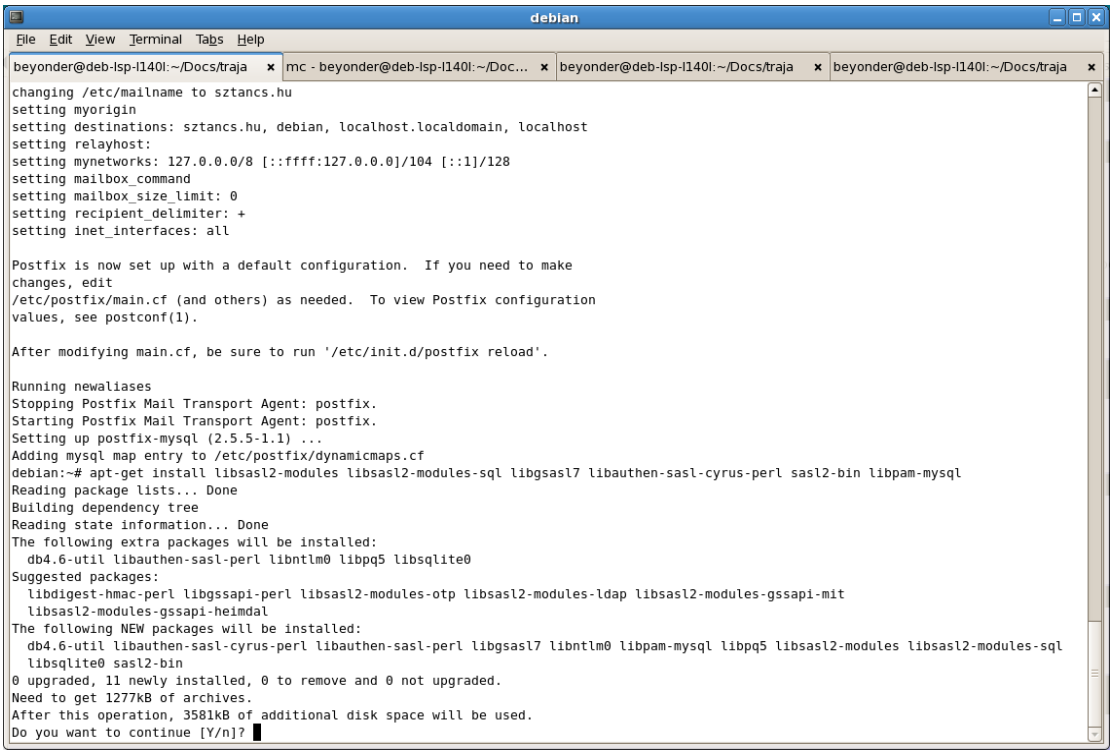




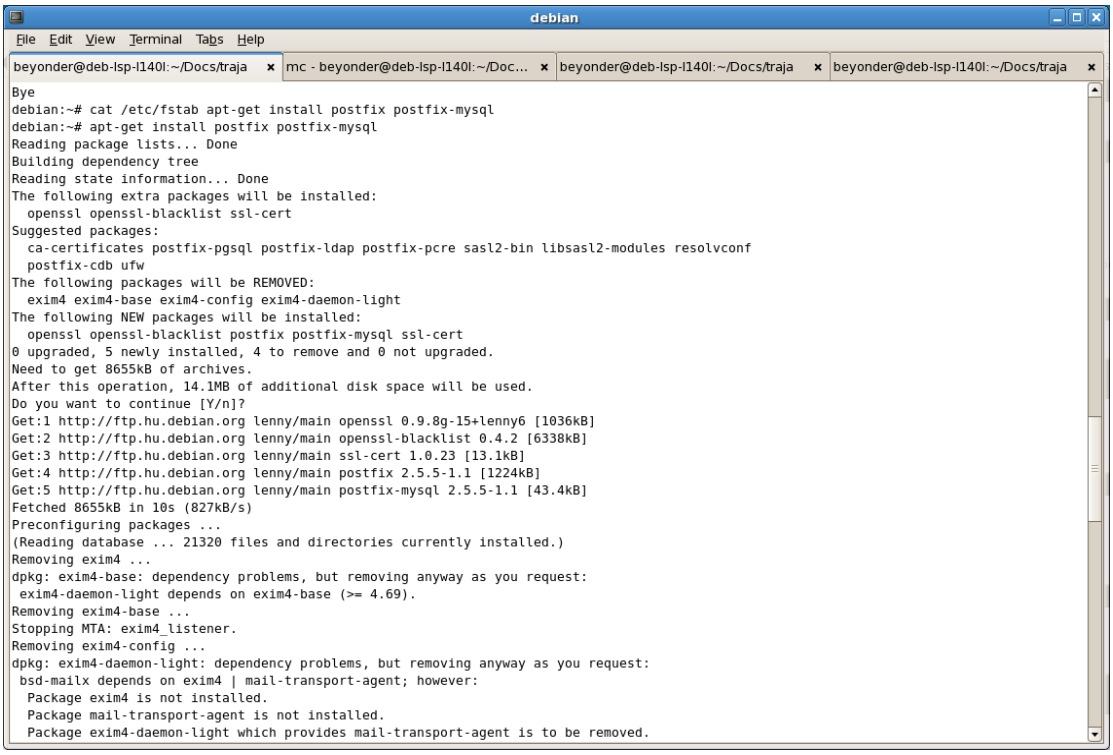
1. ábra: Postfix működési beállítása



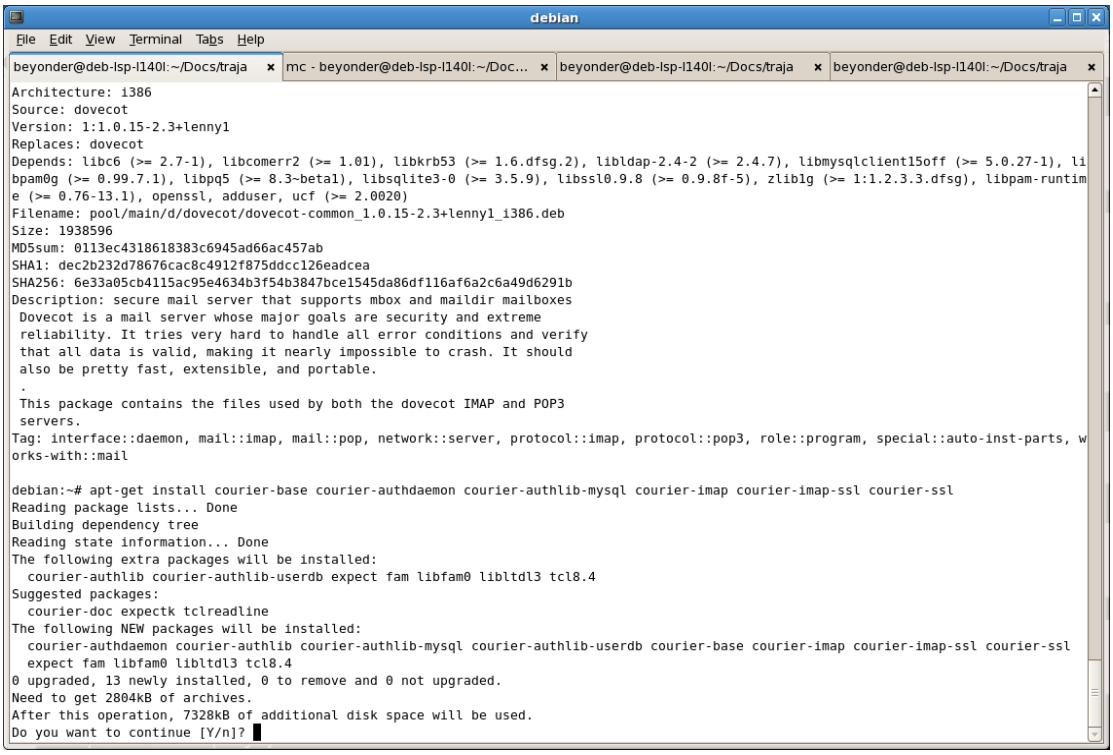
2. ábra: Rendszernév beállítása



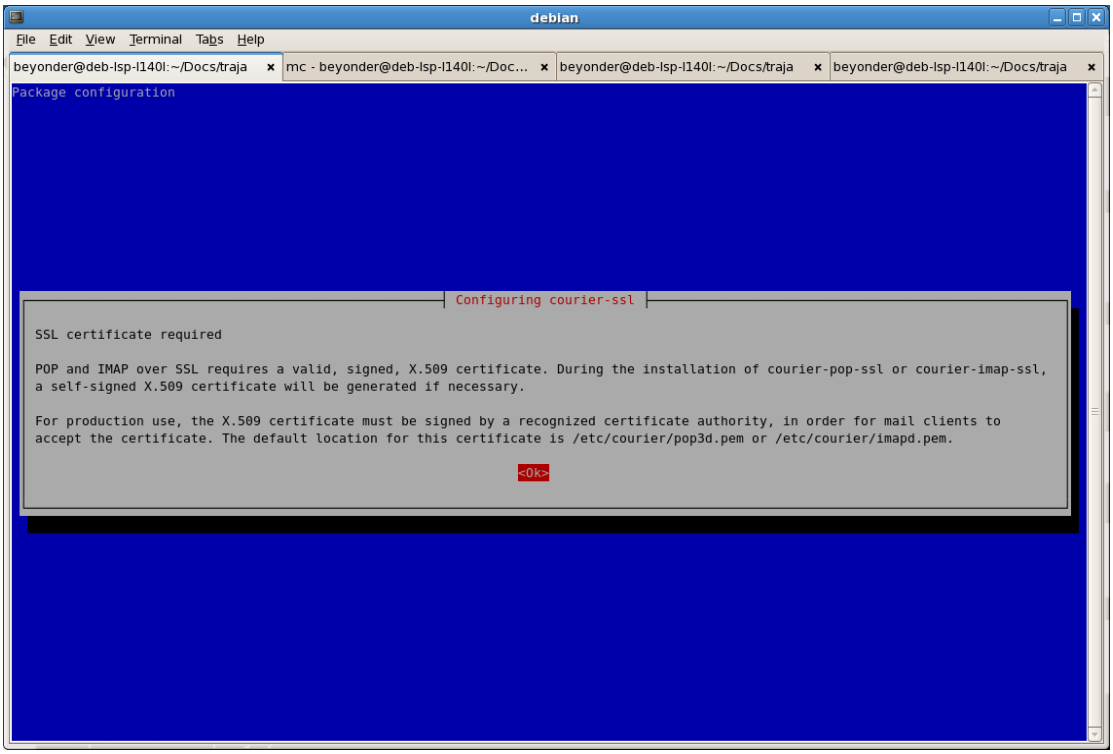
3. ábra: Csomagok és függőségeik telepítése



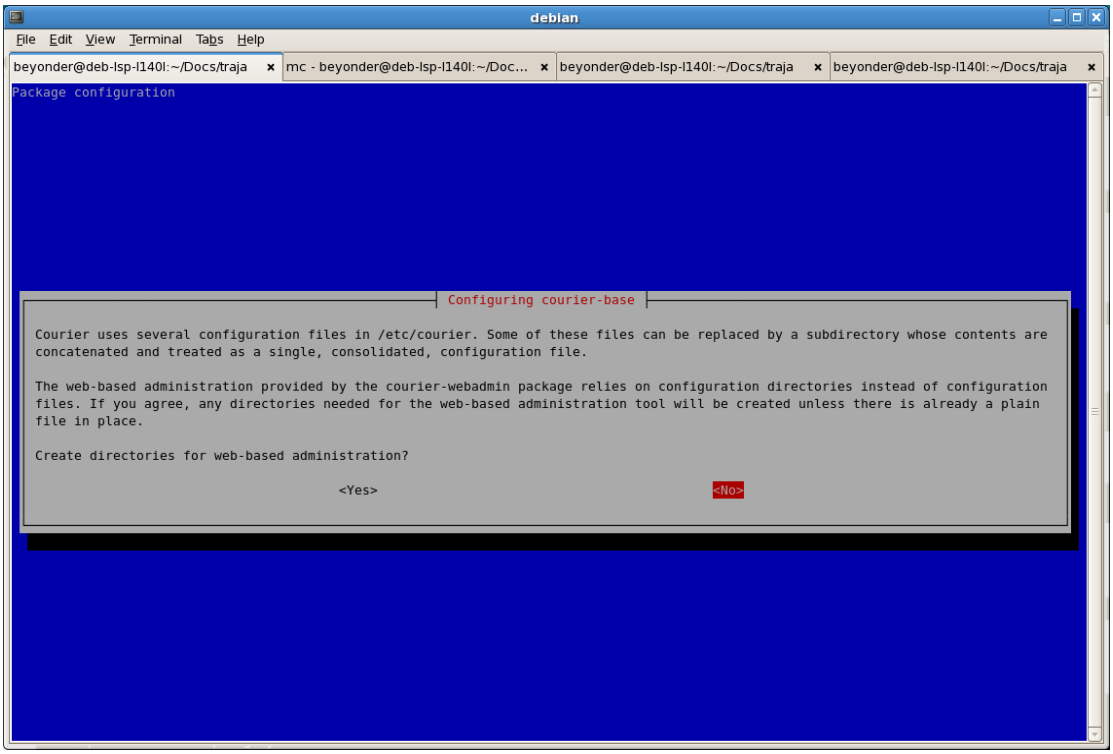
4. ábra: Csomagok és függőségeik telepítése



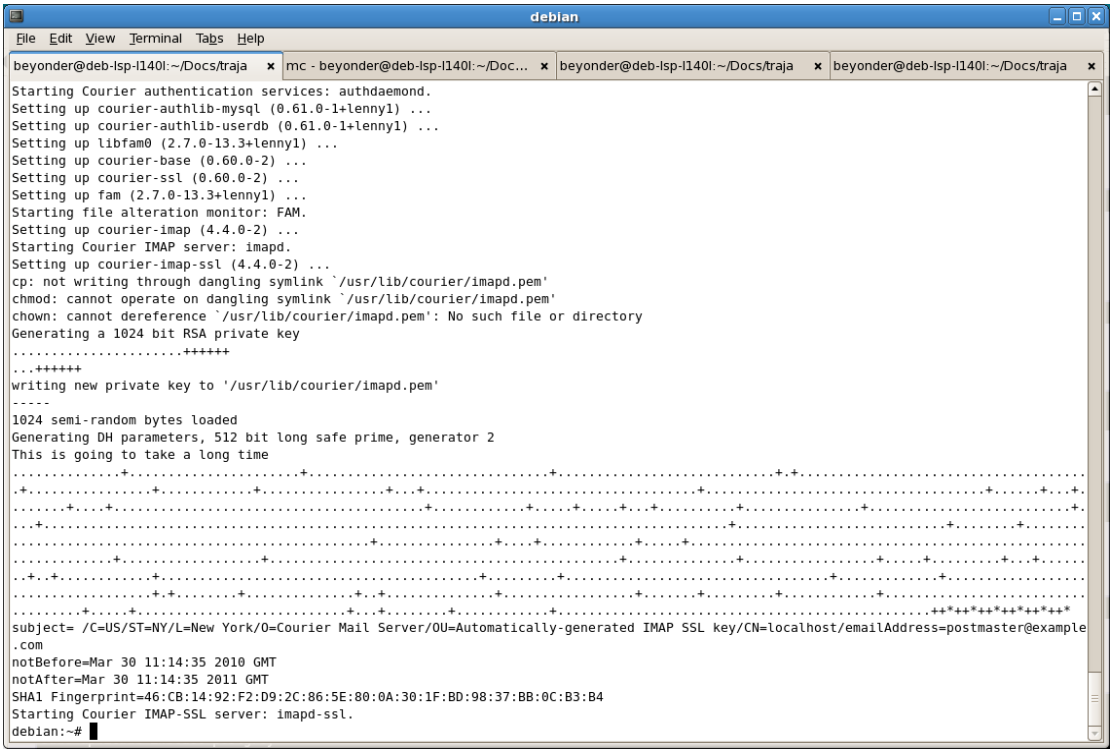
5. ábra: Csomagok és függőségeik telepítése



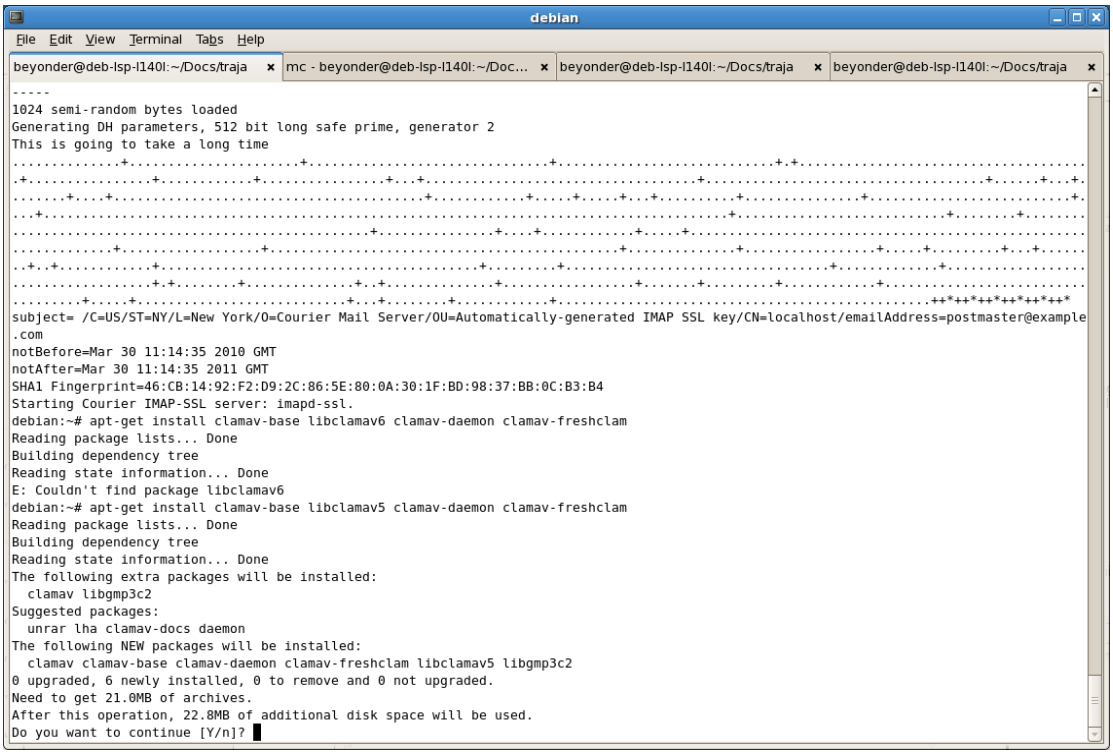
6. ábra: Courier-ssl telepítése



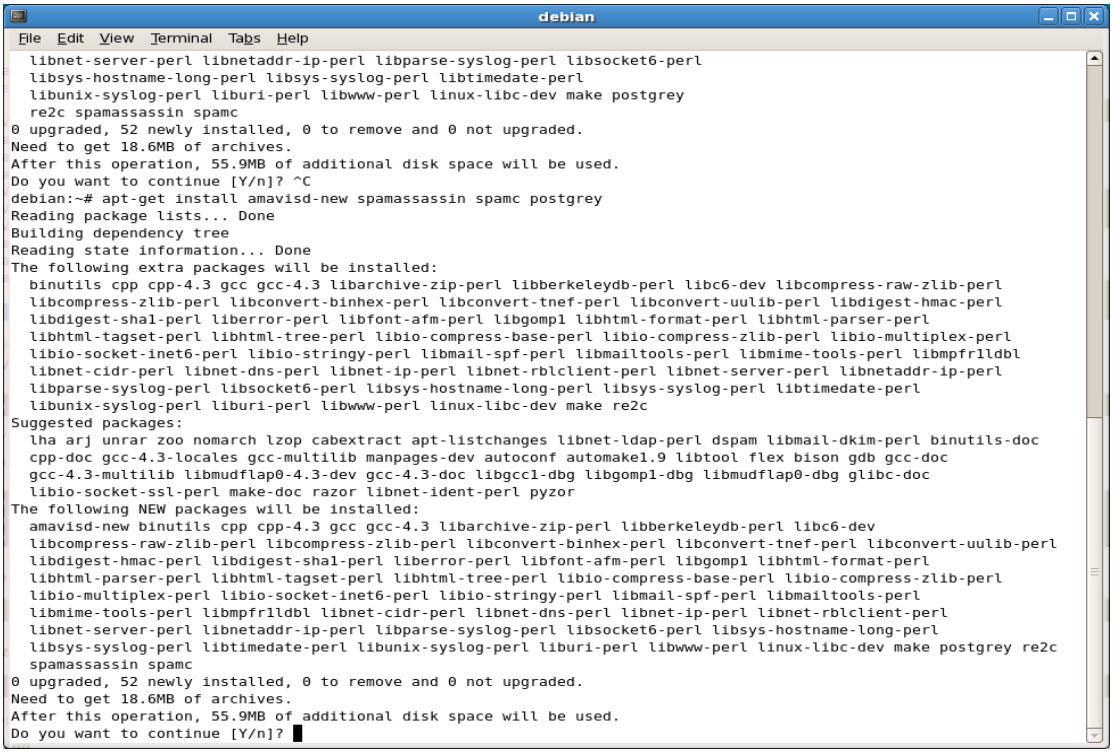
7. ábra: Courier beállítása



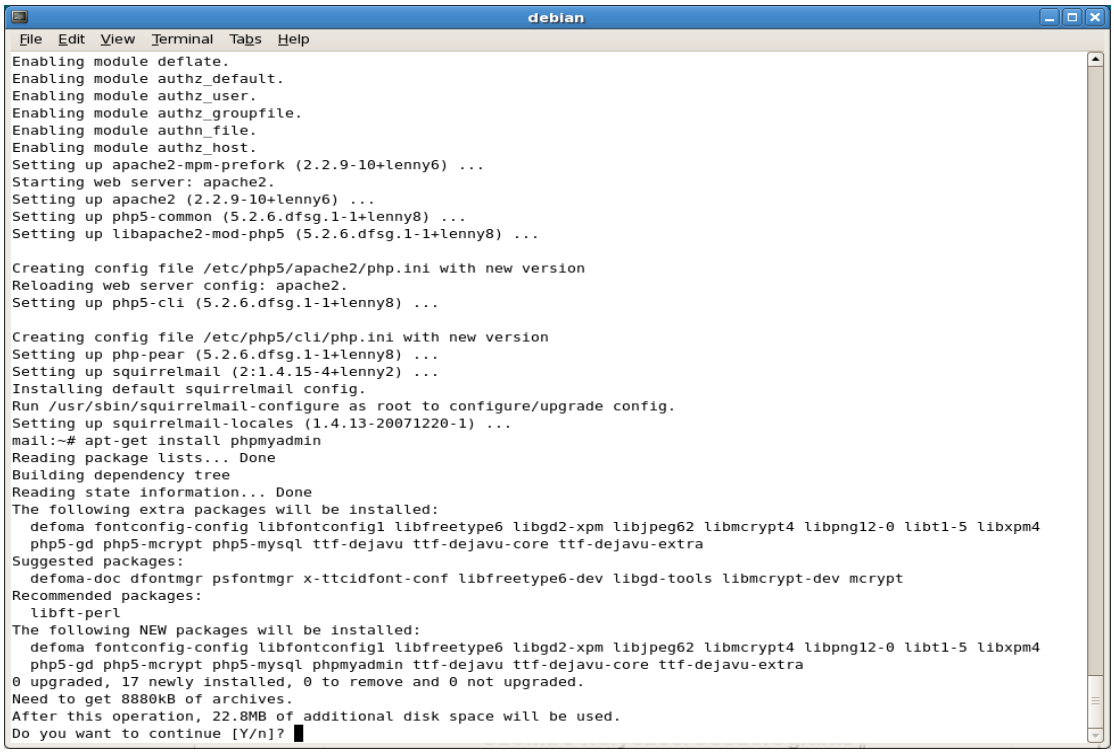
8. ábra: SSL kulcsok elkészítése



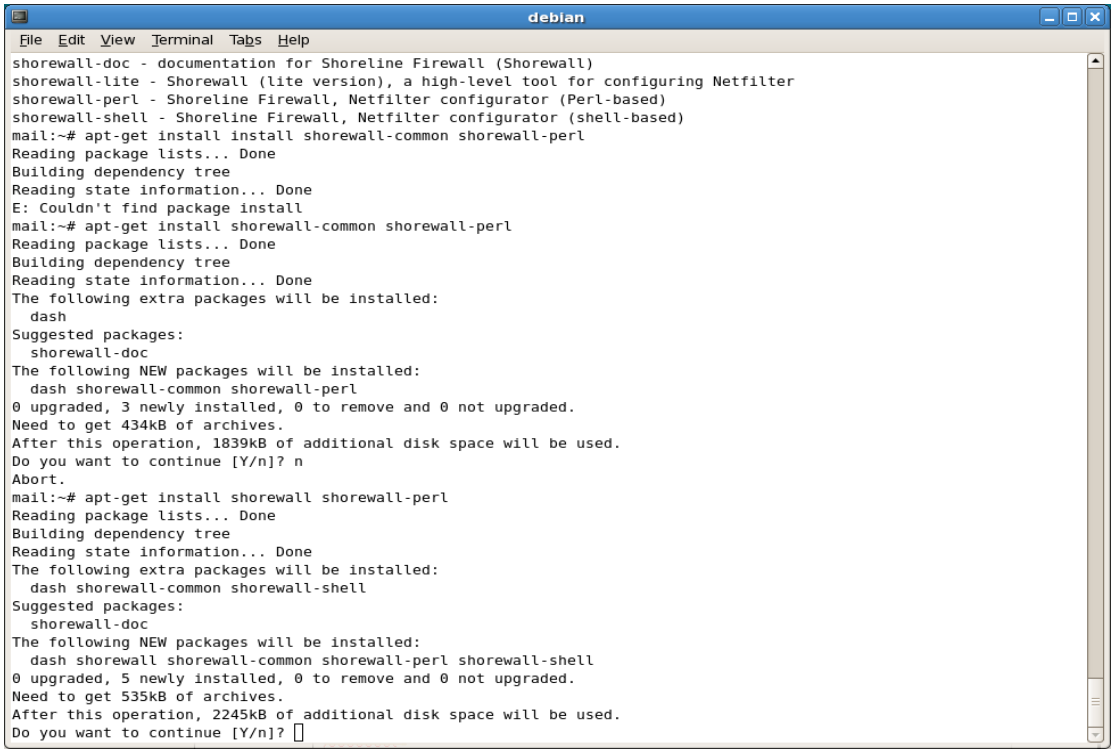
9. ábra: Csomagok és függőségeik telepítése



10. ábra: Csomagok és függőségeik telepítése



11. ábra: Csomagok és függőségeik telepítése



12. ábra: Csomagok és függőségeik telepítése

Szerveren használt csomagok listája

Név	Verzió
acpi-support-base	0.109-11
acpid	1.0.8-1lenny2
adduser	3.110
amavisd-new	1:2.6.1.dfsg-1
apache2	2.2.9-10+lenny6
apache2-mpm-prefork	2.2.9-10+lenny6
apache2-utils	2.2.9-10+lenny6
apache2.2-common	2.2.9-10+lenny6
apt	0.7.20.2+lenny1
apt-utils	0.7.20.2+lenny1
aptitude	0.4.11.11-1
at	3.1.10.2
base-files	5lenny5
base-passwd	2003-05-20
bash	2003-02-04
bash-completion	20080705
bc	1.06.94-3
bind9-host	1:9.5.1.dfsg.P3-1
binutils	2.18.1~cvs2008
bsd-mailx	8.1.2-0.20071201
bsdmainutils	2006-01-10
bsdutils	1:2.13.1.1-1
busybox	1:1.10.2-2
clamav	0.94.dfsg.2-1lenny2
clamav-base	0.94.dfsg.2-1lenny2
clamav-daemon	0.94.dfsg.2-1lenny2
clamav-freshclam	0.94.dfsg.2-1lenny2
console-common	0.7.80
console-data	2:1.07-11
console-tools	1:0.2.3dbs-65.1
coreutils	2006-10-06
courier-authdaemon	0.61.0-1+lenny1
courier-authlib	0.61.0-1+lenny1
courier-authlib-mysql	0.61.0-1+lenny1
courier-authlib-userdb	0.61.0-1+lenny1
courier-base	0.60.0-2
courier-imap	4.4.0-2
courier-imap-ssl	4.4.0-2
courier-ssl	0.60.0-2
cpio	2002-09-13
cpp	4:4.3.2-2
cpp-4.3	4.3.2-1.1

Név	Verzió
cron	3.0pl1-105
dash	0.5.4-12
db4.6-util	4.6.21-11
dc	1.06.94-3
debconf	2001-05-24
debconf-i18n	2001-05-24
debian-archive-keyring	2009-01-31
debian-faq	4.0.4
debianutils	2.30
defoma	0.11.10-0.2
dhcp3-client	3.1.1-6+lenny4
dhcp3-common	3.1.1-6+lenny4
dictionaries-common	0.98.12
diff	2.8.1-12
dmidecode	2002-09-01
dnsutils	1:9.5.1.dfsg.P3-1
doc-debian	4.0.1
doc-linux-text	2008-08-01
dpkg	1.14.29
e2fslibs	1.41.3-1
e2fsprogs	1.41.3-1
ed	2000-07-03
eject	2.1.5+deb1-4
exim4	4.69-9
exim4-base	4.69-9
exim4-config	4.69-9
exim4-daemon-light	4.69-9
expect	5.43.0-17
fam	2.7.0-13.3+lenny1
file	4.26-1
findutils	4.4.0-2
fontconfig-config	2.6.0-3
ftp	0.17-18
gcc	4:4.3.2-2
gcc-4.2-base	4.2.4-6
gcc-4.3	4.3.2-1.1
gcc-4.3-base	4.3.2-1.1
gettext-base	0.17-4
gnupg	1.4.9-3+lenny1
gpgv	1.4.9-3+lenny1
grep	2.5.3~dfsg-6
groff-base	1.18.1.1-21

Név	Verzió
grub	0.97-47lenny2
grub-common	1.96+20080724-16
gzip	1.3.12-6+lenny1
hostname	2.95
iamerican	3.1.20.0-4.4
ibritish	3.1.20.0-4.4
ifupdown	0.6.8+nmu1
info	4.11.dfsg.1-4
initramfs-tools	0.92o
initscripts	2.86.ds1-61
installation-report	2.38
iproute	20080725-2
iptables	1.4.2-6
iputils-ping	3:20071127-1
ispell	3.1.20.0-4.4
kannel	1.4.1-2
kannel-extras	1.4.1-2
klibc-utils	1.5.12-2
laptop-detect	0.13.6
less	418-01-01
libacl1	2.2.47-2
libapache2-mod-php5	5.2.6.dfsg.1-1
libapr1	1.2.12-5+lenny1
libaprutil1	1.2.12+dfsg-8
libarchive-zip-perl	1.18-1
libattr1	1:2.4.43-2
libauthen-sasl-cyrus-perl	0.13-server-3+b1
libauthen-sasl-perl	2002-12-01
libberkeleydb-perl	0.34-1+b1
libbind9-40	1:9.5.1.dfsg.P3-1
libblkid1	1.41.3-1
libbz2-1.0	1.0.5-1
libc6	2.7-18lenny2
libc6-dev	2.7-18lenny2
libc6-i686	2.7-18lenny2
libcap1	1:1.10-14
libcap2	2002-11-02
libclamav5	0.94.dfsg.2-1lenny2
libcomerr2	1.41.3-1
libcompress-raw-zlib-perl	2.012-1lenny1
libcompress-zlib-perl	2.012-1
libconsole	1:0.2.3dbs-65.1
libconvert-binhex-perl	1.119+pristine-3

Név	Verzió
libconvert-tnef-perl	0.17-8
libconvert-uulib-perl	2001-11-01
libcwidget3	0.5.12-4
libdb4.5	4.5.20-13
libdb4.6	4.6.21-11
libdbd-mysql-perl	4.007-1+lenny1
libdbi-perl	1.605-1
libdevmapper1.02.1	2:1.02.27-4
libdigest-hmac-perl	2001-01-07
libdigest-sha1-perl	2.11-2+b1
libdns45	1:9.5.1.dfsg.P3-1
libedit2	2.11~20080614-1
libept0	2000-05-22
liberror-perl	0.17-1
libevent1	1.3e-3
libexpat1	2.0.1-4+lenny3
libfam0	2.7.0-13.3+lenny1
libfont-afm-perl	1.20-1
libfontconfig1	2.6.0-3
libfreetype6	2.3.7-2+lenny1
libgc1c2	1:6.8-1.1
libgcc1	1:4.3.2-1.1
libgcrypt11	1.4.1-1
libgd2-xpm	2.0.36~rc1~dfsg-3+lenny1
libgdbm3	1.8.3-3
libglib2.0-0	2.16.6-3
libglib2.0-data	2.16.6-3
libgmp3c2	2:4.2.2+dfsg-3
libgnutls26	2.4.2-6+lenny2
libgomp1	4.3.2-1.1
libgpg-error0	2001-04-02
libgpm2	1.20.4-3.1
libgsasl7	0.2.26-2
libgssglue1	2000-01-02
libhtml-format-perl	2002-04-02
libhtml-parser-perl	3.56-1+lenny1
libhtml-tagset-perl	3.20-2
libhtml-template-perl	2002-09-01
libhtml-tree-perl	3.23-1
libidn11	1.8+20080606-1
libio-compress-base-perl	2.012-1
libio-compress-zlib-perl	2.012-1
libio-multiplex-perl	2001-09-02

Név	Verzió
libio-socket-inet6-perl	2.54-1
libio-stringy-perl	2.110-4
libisc45	1:9.5.1.dfsg.P3-1
libiscсc40	1:9.5.1.dfsg.P3-1
libisccfg40	1:9.5.1.dfsg.P3-1
libjpeg62	6b-14
libkeyutils1	2001-02-09
libklibc	1.5.12-2
libkrb53	1.6.dfsg.4~beta1-5
libldap-2.4-2	2.4.11-1+lenny1
liblocale-gettext-perl	2001-05-04
liblockfile1	2001-08-03
libltdl3	1.5.26-4+lenny1
liblwres40	1:9.5.1.dfsg.P3-1
libmagic1	4.26-1
libmail-spf-perl	2.005-1
libmailtools-perl	2002-03-01
libmcrypt4	2.5.7-5
libmime-tools-perl	5.427-1
libmpfr1ldbl	2.3.1.dfsg.1-2
libmysqlclient15off	5.0.51a-24+lenny3
libncurses5	5.7+20081213-1
libncursesw5	5.7+20081213-1
libnet-cidr-perl	2000-11-03
libnet-daemon-perl	0.38-1.1
libnet-dns-perl	0.63-2
libnet-ip-perl	1.25-2
libnet-rblclient-perl	2000-05-02
libnet-server-perl	0.97-1
libnetaddr-ip-perl	4.007+dfsg-2+b1
libnewt0.52	0.52.2-11.3+lenny1
libnfsidmap2	0.20-1
libntlm0	0.3.13-1
libpam-modules	1.0.1-5+lenny1
libpam-mysql	0.6.2-1
libpam-runtime	1.0.1-5+lenny1
libpam0g	1.0.1-5+lenny1
libparse-syslog-perl	2001-10-01
libpci3	1:3.0.0-6
libpcre3	7.6-2.1
libplrpc-perl	0.2017-1.1
libpng12-0	1.2.27-2+lenny2
libpopt0	1.14-4
libpq5	8.3.9-0lenny1

Név	Verzió
libreadline5	5.2-3.1
librpcsecgss3	0.18-1
libsasl2-2	2.1.22.dfsg1-23
libsasl2-modules	2.1.22.dfsg1-23
libsasl2-modules-sql	2.1.22.dfsg1-23
libselinux1	2.0.65-5
libsepol1	2.0.30-2
libsigc++-2.0-0c2a	2.0.18-2
libslang2	2.1.3-3
libsocket6-perl	0.20-1
libsqlite0	2.8.17-4
libsqlite3-0	3.5.9-6
libss2	1.41.3-1
libssl0.9.8	0.9.8g-15+lenny6
libstdc++6	4.3.2-1.1
libsys-hostname-long-perl	2001-04-02
libsys-syslog-perl	0.26-1
libt1-5	5.1.2-3
libtasn1-3	2001-04-01
libterm-readkey-perl	2.30-4
libtext-charwidth-perl	0.04-5+b1
libtext-iconv-perl	1.7-1+b1
libtext-wrapi18n-perl	2000-06-06
libtimedate-perl	1.1600-9
libunix-syslog-perl	2001-01-02
liburi-perl	1.35.dfsg.1-1
libusb-0.1-4	2:0.1.12-13
libuuid1	1.41.3-1
libvolume-id0	0.125-7+lenny3
libwrap0	7.6.q-16
libwww-perl	5.813-1
libx11-6	2:1.1.5-2
libx11-data	2:1.1.5-2
libxapian15	1.0.7-4
libxau6	1:1.0.3-3
libxcb-xlib0	1.1-1.2
libxcb1	1.1-1.2
libxdmcp6	1:1.0.2-3
libxext6	2:1.0.4-1
libxml2	2.6.32.dfsg-5
libxmuu1	2:1.0.4-1
libxpm4	1:3.5.7-1
linux-image-2.6-686	2.6.26+17+lenny1

Név	Verzió
linux-image-2.6.26-2-686	2.6.26-21lenny4
linux-libc-dev	2.6.26-21lenny4
locales	2.7-18lenny2
lockfile-progs	0.1.11-0.1
login	1:4.1.1-6+lenny1
logrotate	3.7.1-5
lsb-base	2003-02-20
lsof	4.78.dfsg.1-4
lzma	4.43-14
m4	1.4.11-1
make	3.81-5
makedev	2.3.1-88
man-db	2.5.2-4
manpages	2003-05-01
mawk	1.3.3-11.1
mc	2:4.6.2~git200803
mime-support	3.44-1
mktemp	2001-05-09
mlocate	0.21.1-1
module-init-tools	2003-04-01
mount	2.13.1.1-1
mutt	1.5.18-6
mysql-client	5.0.51a-24+lenny3
mysql-client-5.0	5.0.51a-24+lenny3
mysql-common	5.0.51a-24+lenny3
mysql-server	5.0.51a-24+lenny3
mysql-server-5.0	5.0.51a-24+lenny3
nano	2.0.7-4
ncurses-base	5.7+20081213-1
ncurses-bin	5.7+20081213-1
ncurses-term	5.7+20081213-1
net-tools	1.60-22
netbase	4.34
netcat-traditional	1.10-38
nfs-common	1:1.1.2-6lenny1
ntpdate	1:4.2.4p4+dfsg-8
openbsd-inetd	0.20080125-2
openssh-blacklist	2000-04-01
openssh-blacklist-extra	2000-04-01
openssh-client	1:5.1p1-5
openssh-server	1:5.1p1-5
openssl	0.9.8g-15+lenny6
openssl-blacklist	2000-04-02

Név	Verzió
passwd	1:4.1.1-6+lenny1
patch	2.5.9-5
pciutils	1:3.0.0-6
perl	5.10.0-19lenny2
perl-base	5.10.0-19lenny2
perl-modules	5.10.0-19lenny2
php-pear	5.2.6.dfsg.1-1
php5-cli	5.2.6.dfsg.1-1
php5-common	5.2.6.dfsg.1-1
php5-gd	5.2.6.dfsg.1-1
php5-mcrypt	5.2.6.dfsg.1-1
php5-mysql	5.2.6.dfsg.1-1
phpmyadmin	4:2.11.8.1-5+lenny3
portmap	6.0-9
postfix	2.5.5-1.1
postfix-mysql	2.5.5-1.1
postgrey	1.31-3.2
procmail	3.22-16
procps	1:3.2.7-11
psmisc	2022-06-01
python	2.5.2-3
python-central	2000-06-08
python-minimal	2.5.2-3
python2.5	2.5.2-15+lenny1
python2.5-minimal	2.5.2-15+lenny1
re2c	0.13.5-1
readline-common	5.2-3.1
reportbug	3.48
rsyslog	3.18.6-4
sasl2-bin	2.1.22.dfsg1-23
sed	4.1.5-6
shorewall	4.0.15-1
shorewall-common	4.0.15-1
shorewall-perl	4.0.15-1
shorewall-shell	4.0.15-1
spamassassin	3.2.5-2+lenny2
spamc	3.2.5-2+lenny2
squirrelmail	2:1.4.15-4+lenny2
squirrelmail-locales	1.4.13-20071220-1
ssl-cert	1.0.23
sysv-rc	2.86.ds1-61
sysvinit	2.86.ds1-61
sysvinit-utils	2.86.ds1-61
tar	1.20-1

Név	Verzió
tasksel	2.78
tasksel-data	2.78
tcl8.4	8.4.19-2
tcpd	7.6.q-16
telnet	0.17-36
texinfo	4.11.dfsg.1-4
time	2001-07-23
traceroute	2.0.11-2
ttf-dejavu	2.25-3
ttf-dejavu-core	2.25-3
ttf-dejavu-extra	2.25-3
tzdata	2010a-0lenny1
ucf	3.0016
udev	0.125-7+lenny3
update-inetd	4.31
usbutils	0.73-10lenny1
util-linux	2.13.1.1-1
vim-common	1:7.1.314-3+lenny2
vim-tiny	1:7.1.314-3+lenny2
w3m	0.5.2-2+b1
wamerican	2006-02-03
wget	1.11.4-2+lenny1
whiptail	0.52.2-11.3+lenny1
whois	2004-07-30
x11-common	1:7.3+20
xauth	1:1.0.3-2
zlib1g	1:1.2.3.3.dfsg-12

9 Irodalomjegyzék

1: Debian ingyenes operációs rendszer,
http://www.debian.org/intro/free

2: Linux kernel,
http://www.kernel.org/

3: GNU project,
http://www.gnu.org/

4: MySQL – Wikipédia, Wikipédia, 2010.04.23.,
http://hu.wikipedia.org/wiki/MySQL

5: MySQL adatbázis,
http://www.mysql.com/

6: MTA, Wikipedia, 2010 03.28.,
http://en.wikipedia.org/wiki/Message_transfer_agent

7: Mail exchanger, Wikipedia, 2010.04.24.,
http://en.wikipedia.org/wiki/MX_record

8: Postfix,
http://www.postfix.org/

9: FQDN, Wikipedia, 2010.04.22.,
http://hu.wikipedia.org/wiki/FQDN

10: Courier,
http://www.courier-mta.org/

11: SSL, Wikipedia, 2005.06.20.,
http://wiki.hup.hu/index.php/SSL

12: Spam, Wikipedia, 2010.04.29.,
http://en.wikipedia.org/wiki/Spam_(electronic)

13: Amavisd-new,
http://www.ijs.si/software/amavisd/

14: Squirrelmail,
http://squirrelmail.org/

15: phpMyAdmin,
http://www.phpmyadmin.net/

16: Shorewall,
http://www.shorewall.net/

17: SMTP, Wikipedia, 2010.04.11.,
http://en.wikipedia.org/wiki/Simple_Mail_Transfer_Protocol

18: Catch-all, Wikipedia, 2010.01.15.,
http://en.wikipedia.org/wiki/Catch-all
19: Chroot, Wikipedia, 2007.02 23.,
http://wiki.hup.hu/index.php/Chroot