

Biztonságos Web-szerver kialakítása
Debian GNU/Linux 2.2 rendszeren
v1.0.2

Zákány Gergely
narancs1@externet.hu

Minden jog fenntartva a szerző számára (c) 2000 Zákány Gergely.

Engedélyezett ennek a dokumentumnak a lemásolása, terjesztése és / vagy módosítása az FSF/GNU Free Documentation License (Szabad Dokumentációs Licensze) 1.1-es vagy újabb verziója alatt. (Ezt a licenszet itt is olvashatjuk: <http://www.gnu.org/copyleft/fdl.html>) Nincs eltérő szekció (*invariant section*). Nincs első és hatsó borító szöveg sem. A licensz a "GNU Free Documentation License" című függelékben olvasható.

Tartalomjegyzék

TARTALOMJEGYZÉK	5
ÁBRAJEGYZÉK	8
ELŐSZÓ	9
KÖSZÖNETNYILVÁNÍTÁS	10
I. BEVEZETÉS	11
Bevezetés üzleti szempontból	11
Bevezetés gyakorlati szempontból	14
II. ALAPFOGALMAK.....	18
1. A GNU PROJEKT, A GPL LICENSZ	19
2. A LINUX KERNEL	21
Néhány technikai jellegű információ	22
Nemzetközi változat	23
3. A DEBIAN PROJEKT	24
Mire jó egy terjesztés?	24
4. AZ APACHE PROJEKT, AZ APACHE MODULJAI	27
5. A WEB-SZERVER HELYE A HÁLÓZATBAN (INTERNET/INTRANET).....	31
6. A WEB-SZEMÉLYZET FELÉPÍTÉSE	35
7. WEB-PROXY FOGALMA ÉS HELYE A HÁLÓZATBAN	36
8. BIZTONSÁGI ALAPOK (HARDVER, SZOFTVER).....	36
8.1 Általános irányelvek	37
8.2 A Linux kernel biztonságát növelő projektek.....	41
8.3 A Web-alkalmazások biztonsága	43
9. SSH - TÁVOLI MENEDZSMENT	44
10. PHP3 ALAPOK (DINAMIKUS WEBLAP-KÉSZÍTÉS)	45
11. MYSQL ALAPOK (ADATBÁZIS SZERVER)	49
III. TERVEZÉS.....	52
1. A FELADAT FELMÉRÉSE - SKÁLÁZHATÓSÁG, ALTERNATÍVÁK, HARDVER.....	52
2. KÖLTSÉGEK BECSLÉSE MINTAPÉLDA ALAPJÁN.....	53
3. BIZTONSÁG	55
3.1 Milyen programok lehetnek / nem lehetnek egy Web-szerveren?	55
3.2 A partíciók megtervezése általánosan és a mintapéldához.....	57
3.3 A biztonsági mentés (backup) lehetőségei, módszerei és javasolt paraméterei.....	60
3.4 Szoftveres UPS (szünetmentes tápegység) felügyelet soros porton keresztül	63
3.5 A szükséges felhasználók/csoportok és a lemezkvóta megtervezése	65
IV. MEGVALÓSÍTÁS.....	68
1. GYORSTALPALÁS	68
1.1 A szoftver beszerzése: CD-set, vagy FTP tükör.....	68
1.2 A telepítés menete	70
1.2.1 Indítás CD-ről vagy floppy-ról.....	70
1.2.2 Szükséges alapbeállítások, particionálás	71

1.2.3 A hálózat beállítása	73
1.2.4 Alaprendszer telepítése, újraindítás a merevlemezről	75
1.2.5 A jelszórendszer beállítása. („MD5”, „Shadow password”).....	75
1.2.6 Az „apt” program beállítása	76
1.2.7 A „dselect” program.....	78
A csomagok kiválasztása	79
1.2.8 A feltelepített programok konfigurálása	81
2. FINOMÍTÁS.....	84
2.1 Első lépések.....	84
2.2 Az <i>Inetd.conf</i> finomhangolása – a nem biztonságos szolgáltatások letiltása.....	88
2.3 A levelező démon beállítása	89
A titkosítás beállítása	91
2.3 Másik gép használata a fordításokhoz, miért?	94
2.4 Személyre szabott kernel konfigurálása és fordítása kézzel és a „kernel-package” csomaggal. A „lilo” beállításai.....	94
2.3 Az Apache finomhangolása, esetleges újrafordítása hardver és feladatorientáltan.....	102
Az Apache fordítása.....	109
2.4 Az SSH konfigurációjának finomhangolása	111
2.6 Szoftveres figyelő („watchdog”) beállítása.....	113
2.7 E-mail titkosító kulcspárok létrehozása a „gpg” programmal	114
2.8 A rendszernapló (log) és kezelése.....	116
2.8.1 A „syslog” démon kiválasztása	117
2.8.2 A naplófájlok rotálásának beállítása.....	117
2.8.3 A Web-szerver naplófájljai	118
2.8.4 A napló automatikus ellenőrzése.....	119
2.9 A Web-szerver statisztikái.....	120
2.10 Az „upsd” beállítása	121
2.11 A biztonsági mentés időzítése	122
2.12 A szükséges felhasználók/csoportok létrehozása és a lemezkvóták beállítása.....	122
2.13 A „/etc/fstab” és az „init script”-ek beállítása.....	123
2.14 A „tripwire” program beállítása és használata	126
V. EGY GYAKORLATI PÉLDA BEMUTATÁSA.....	129
VI. A JÖVŐ.....	133
1. AZ ÚJ KERNEL ÉS A KHTTPD	133
2. AZ ÚJ DEBIAN	136
3. AZ ÚJ APACHE	136
4. PHP4	136
VII. ALTERNATÍVÁT NYÚJTÓ PROGRAMOK A DEBIAN-BAN	138
1. ALTERNATÍVÁK A HTTPD-RE	138
1.1 Roxen.....	138
1.2 Zope – Z Object Publishing Enviroment.....	139
1.3 Kisebb szerverek.....	139
2. ALTERNATÍVÁK A DINAMIKUS HTML-EK GENERÁLÁSÁRA	140
3. ALTERNATÍVÁK SQL SZERVERRE.....	140
4. ALTERNATÍVÁK A TÁVOLI BEJELENTKEZÉSRE.....	141
5. ALTERNATÍVÁK AZ EGYÉB PROGRAMOKRA	142
VIII. ÖSSZEGZÉS.....	143

IRODALOMJEGYZÉK.....	144
FÜGGELÉK.....	145
1. A GPL v2 LICENSZ MAGYAR NYELVŰ FORDÍTÁSA.....	145
2. A BSD LICENSZ	149
3. A DEBIAN „SOCIAL CONTRACT” (TÁRSADALMI SZERZŐDÉS) / DFSG.....	150
4. DEBIAN INFORMÁCIÓK	152
5. RÖVIDÍTÉSEK, SZAKSZAVAK JEGYZÉKE	152
6. AJÁNLOTT RFC-K	153
7. A LIDS RENDSZER BEÁLLÍTÁSA	154
8. A „GNU FREE DOCUMENTATION LICENSE”	156
MELLÉKLET	161

Ábrajegyzék

1. kép - A Web szerver helye a hálózatban	31
2. kép - Üdvözlőkép	70
3. kép - Speciális indítási paraméterek	71
4. kép - Indítási metódusok	71
5. kép - Merevlemez partícionálás	72
6. kép - A cfdisk program	72
7. kép - Modulok kiválasztása és betöltése a modconf programmal	73
8. kép - Hálózati modulok tallózása	74
9. kép - Az APT program beállítása	76
10. kép - dselect - Főmenü	78
11. kép - dselect - Súlyó	79
12. kép - dselect - Csomaglista	80
13. kép - a "snort" program beállítása	83
14. kép - Postfix – igazolás készítése a CA.pl programmal	93
15. kép - make menuconfig	97
16. kép - make xconfig	97
17. kép - rules fájl szerkesztése	111
18. kép - Webalizer statisztika	121
1. táblázat - Az Apache moduljai	30
2. táblázat - Az engedélyezett szolgáltatások listája	32
3. táblázat - PHP3 csomagok a Debian-ban	47
4. táblázat - MySQL csomagok a Debian-ban	51
5. táblázat - Partíció-terv	57
6. táblázat - Felhasználók listája 1.	66
7. táblázat - Felhasználók listája 2.	66
8. táblázat - Felhasználók listája 3.	66
9. táblázat - A kernel részei	95
10. táblázat - Autentikációs modulok	107
11. táblázat - PostgreSQL csomagok a Debian-ban	141
12. táblázat - Alternatív csomagok távoli bejelentkezésre a Debian-ban	142
13. táblázat - Shell-ek a Debian-ban	142

Előszó

E diplomamunka remélhetőleg nem csupán számomra lesz hasznos, mint életem egy mérföldköve, hanem a magyar számítógép használók / rendszergazdák is hasznos segédeszközként forgathatják majd, amíg (és milyen gyorsan) el nem avul a benne lévő információ.

Amióta figyelemmel kísérem a Linux kernelre¹ épülő operációs rendszerek fejlődését, mind az informatika (hardver), mind maga a Linux és a köréje épülő programok óriási fejlődésen mentek keresztül. Persze, ha mindezen fejlődést Richard M. Stallmann² szemszögéből vizsgálom – az ő több évtizedes szakmai tapasztalatai szerint – akkor még hangsúlyosabban érezhetővé válna számomra, hogy exponenciális fejlődés van folyamatban mind a két szektorban³. A Linux úgyszólván a semmiből jött. Egy finn egyetemista programozónak⁴ nem tetszett az otthoni PC-jén lévő operációs rendszer és elkezdett írni egy rendszermagot saját szórakozására. Később a különböző szabad szoftvereket kezdték el párosítani ezzel a kernellel (mely szintén szabad szoftver). Sikerrel. Ma már széles körben elterjedtek (és egyre jobban elterjednek) ezek a rendszerek, világcégek támogatják fejlődését, miközben kezd jó üzletté válni használata, forgalmazása és kereskedelmi szoftverek e platformra ültetése.

Hogy hasznos-e, szükséges-e vagy sem az emberiség szempontjából ez a rohanó fejlődés, azt ilyen távlatokból helyesen még nem szemlélhetjük. Vajon hova vezet ez az őrült tempó és a „Bigger, Better, Faster, More” (nagyobb, jobb, gyorsabb, több) szemlélete, hosszabb távon nem káros-e a társadalomra, az emberiségre? A hatalom azok kezébe kerül, akik lehető leghamarabb birtokolják az információt. A hadviselés is már a kiber-háborúban, az ellenség informatikai eszközeinek megbénításában gondolkodik.

Nem tagadható, a XXI. század az információs társadalom kora, s „aki kimarad, lemarad”. Egy biztos: egyre nehezebb befogadni és feldolgozni az egyre nagyobb mennyiségű információt. Ezért remélem, hogy a magyar nyelvű olvasók számára ez a mű hasznos, friss és a gyakorlatban felhasználható információkat nyújthat.

¹ Kernel: az operációs rendszer magja

² A szabad szoftver mozgalom egyik alapítója, részletesebben később.

³ Hardver / szoftver

⁴ Linus Torvalds

Köszönetnyilvánítás

Köszönettel tartozom a következő embereknek:

- Szüleimnek, amiért lehetővé tették, hogy tanulhassak.
- Konzulensemnek, Borbély Istvánnak az önzetlen segítségért.
- Lektoraimnak: Bíró Dávidnak, Nagy Attilának, Sári Gábornak és a többieknek.
- A szolnoki Linux-klub tagjainak, főképp: Böszörményi Zoltánnak, Herczeg Ferencnek, Kerekes Gyulának, Sári Gábornak és Takács Sándornak és a szakmai segítségért.
- A magyar Debian fejlesztőknek, kiemelten: Madarász Gergelynek, Szalay Attilának.
- A Security-I levelező lista tagjainak, kiemelten: Magosányi Árpádnak, Mátó Péternek.
- Scheidler Balázsnak a `syslog-ng` programért.
- Az egész magyar Linux-os közösségnek, az LME⁵-nek, az MLF⁶-nek, a levelezőlistáknak. A Linux-os konferenciák és összejövetelek szervezőinek.
- Tanárimnak.

Továbbá köszönet illeti magukat a programozókat, akik a Debian GNU/Linux 2.2 rendszerben lévő összes programot szabadidejükben fejlesztették, tesztelték, hogy számunkra, a felhasználóknak elérhetővé váljanak.

⁵ Linux-felhasználók Magyarországi Egyesülete

⁶ Magyar Linux Felhasználók egyesülete

I. Bevezetés

Ez a munka, lévén gyakorlati jellegű, nem igazán az üzleti világban és pénzügyekben jártas menedzserekhez szól, de nem is a kezdő számítógép-felhasználókhoz. A dolgozat feltételezi, hogy az olvasó rendelkezik a megfelelő számítástechnikai / informatikai / hálózati alapismeretekkel. Megfelelő információ-háttér hiány esetében javaslom az irodalomjegyzékben lévő megfelelő anyagok átolvasását és a megjelölt Internet-címek felkeresését.

Bár maga a munka szakmai szempontból viszonylag szűk rétegnek szól, úgy gondoltam, hogy legyen ezen dolgozatnak két bevezetése. Az egyik szóljon az üzleti, a másik pedig a gyakorlati / technikai beállítottságú embereknek. Az első a döntéshozóknak, a második a megvalósítóknak. A teljes dolgozat inkább a második csoportnak segít, (bár az első csoport tagjai is elolvashatják, ha akad rá idejük) akik ezekkel a felvázolt problémákkal nap mint nap találkozhatnak és a problémák megoldása az ő hatás- / feladatkörükbe tartozik. Az első csoport tagjainak a második csoport tagjaival való sikeres együttműködés eléréséhez ajánlom olvasásra a *II. Alapfogalmak*, *III. Tervezés*, és a *V. Egy gyakorlati példa bemutatása* című fejezeteket.⁷ Mivel a döntéshozónak ott kell lennie a tervezési fázis bizonyos részeinél, szükséges, hogy ismerje az alapfogalmakat és az alapvető problémákat, hogy birtokában legyen az információknak, hogy józanul, részlelhajlás és előítéletek nélkül tudjon dönteni. A másik általam kiemelt fejezetben példákat sorolok fel többek között arról is, hogy milyen minimális (értsd: olcsó, meglévő, elérhető) hardverkörnyezetben fut és működőképes a leírt rendszer.

Habár a konkrét példa egy kisebb cég feladatmegoldásának példáját feszegeti, ez a rendszer mind oktatási, mind költségvetési intézményekben használható hasonló, (vagy teljesen más) célokra is. Csupán azért használom a kiscég-példát, mert pl. a főiskolákon és az Internet-szolgáltatóknál már széleskörűen használják, nem egy helyen szerver célokra kizárólagosan is.

Bevezetés üzleti szempontból

Napjaink egyre gyorsuló és élesedő üzleti versenye megkívánja a közepes és kisvállalatok, társaságok megjelenését az Interneten, az elektronikus üzletben. Ennek alapvető feltétele, hogy az adott cég naprakész információkat tegyen közzé magáról és termékeiről, szolgáltatásairól egy minden érdeklődő által elérhető helyen. Megoldás lehet az Internet, amely dinamikus fejlődésével és viszonylag állandó elérhetőségével és rendelkezésre-állásával fellendítheti és felgyorsíthatja a kapcsolatfelvételt a cégek

⁷ Ajánlom továbbá a szabad szoftver licensek elolvasását a függelékben.

között, a cég-fogyasztó kommunikációt, az esetleges elektronikus úton való értékesítést. Tehát az adott cégnek egy Web-lapot (vagy inkább komplex Web-helyet) kell készíteni (vagy inkább készíttetnie egy erre szakosodott céggel) és azt elérhetővé tenni az Interneten. Ez megoldható úgy is, hogy a cég mások szolgáltatásait igénybe véve „kibérel” egy adott méretű helyet egy már működő, Internetre kapcsolt számítógépen. Ez esetben fizetnie kell a szolgáltatás havidíját, amely a Web-hely méretétől és a generált hálózati forgalomtól egyaránt függ. Nem szeretnék konkrét számokban és szolgáltató cégek ajánlataiban tallózni, hiszen ezek olyan dinamikusan változnak a piaci versenyben, hogy mire e munka elkészül, már rég érvényét veszti az információ. Mindenesetre ez a szolgáltatás számunkra hosszabb távon elég költséges lehet. Kifejezetten rossz megoldás is lehet ez a piaci verseny miatti változások gyorsasága miatt: lehet, hogy az adott szolgáltató tönkremegy, átalakul vagy felvásárolják. Ezek tehát mind kockázati tényezők. Ekkor váltanunk kell, és valószínű, hogy az új cégnél nem úgy fog működni a drágán elkészített Web-helyünk, ahogy régen, stb.

Egy köztes megoldást is számba vehetünk. A példabeli cégünk vesz egy kiszolgáló számítógépet, melyet az Internetre köttet a szolgáltató számítógépközpontjában. Ekkor viszont nincs teljes kontrollunk a hardverhez és a mentésekhez. Ez a rendszer biztonságát és hitelességét nagyban veszélyeztetheti.

A harmadik megoldás esetében a szervergép a cégünk telephelyén van. Ekkor az Internet-kapcsolat díja lesz a fix havi költség, melyet forgalom és / vagy idő után számláz a szolgáltató. Több nagyobb neves hardver / szoftver gyártó cég is kínál lehetőségeket, komplett termékeket erre a problémára / szituációra. Ezek természetesen szintén igen költségesek és nagy kezdeti befektetést igényelnek. Mivel a mi kis cégünknek úgyis csak nyűg ez az Internetes mizéria, ráadásul a munkatársak sem értenek az egészhez megfelelő színvonalon, ezért – ha már van – akkor a cég rendszergazdáját kéri meg egy gazdaságosabb megoldás keresésére. (Hiszen lehet, hogy a hozzá nem értés miatt az egész befektetés csak kidobott pénz lenne.) Sajnos a döntéshozók sokszor azt hiszik, hogy az informatikában is úgy működik a rendszer, mint másutt: minél többet fizet az ember, annál jobb terméket kap cserébe. Amennyiben nincs rendszergazdája és számítógépes hálózata a cégnek, akkor vagy alkalmaz (pl. megbízással szerződéssel) egy rendszergazdát, vagy inkább az első verziót választja és az egész problémát ráhagyja a szolgáltatóra.

A másik fő probléma a tartalom naprakészen tartása. Ez ma már elképzelhetetlen „statikus” Web-lapokkal. Ha pl. a cég az árlistáját is közzé szeretné tenni a potenciális vásárlóknak, amely hetente változik, bizony sokba kerülne minden héten átíratni az oldalt. Mindezt ma már rábízhatjuk a kiszolgálón lévő programokra, melyek egy adatbázisban tárolják az adatokat és abból készítik el a Web-lap adott részeit. A cég

munkatársainak semmi mást nem kell tenni, csupán ezeket az adatokat frissíteni, a program a többit elvégzi helyettük.

Tehát a cégünknek a harmadik esetben vennie kell egy kiszolgáló gépet, egy Internetes előfizetést, szerver operációs rendszert, Web-szerver programot, amely képes dinamikus Web-lapok generálására adatbázisból, továbbá egy adatbázis-szerver programot és alkalmazni (legalább) egy rendszergazdát. Ha körülnézünk a kereskedelmi hardver / szoftver piacon és a kész megoldások árait megnézzük, kiderül egyhamar, hogy a mi kis cégünk nem bír el egy ilyen hosszú távon megtérülő, nagy beruházást. Ekkor jön a házi barkácsolás és a részenkénti megvásárlás, de ez se mindig költségkímélő, hiszen, még ha össze is szereltetünk pár százezerért egy kiszolgálónak szánt gépet, ha még lenne is pénz az Internet-előfizetésre, már biztos, hogy a szoftvereket nem tudjuk megvenni (jogszerűen használni).

Ekkor jön a képbe a szabad szoftver. Mi? – merülhet fel a kérdés a kedves olvasóban. Szabad, bárki által ingyenesen használható, terjeszthető, másolható, nyitott forráskódú és bizonyos feltételek között módosítható szoftver. Tehát a tervezett fejlesztés szoftver oldalának költsége cégünknel **0 Ft**. Erre van szükségünk.

Ez a könyv azt mutatja be, hogyan lehet egy ilyen kiszolgálót adott hardver és hálózati körülmények esetén szabad szoftverekkel installálni, beállítani, saját igényeinkre szabni, működtetni, karbantartani – mindezeket a biztonság erős kihangsúlyozásával. Vagyis a döntéshozó (felhasználó) a rendszergazdával együtt az egész rendszer hardver (és hálózat), szoftver (felhasználási struktúrával) tervezési folyamatát közösen megbeszélve egy költségkímélő, jól és biztonságosan működő rendszert hozhat létre. Ez mindenképp előnyére válik a cégnek az elektronikus kereskedelembe (és az információs hálózatba) való bekapcsolódáskor.

Üzleti szempontból a kitűzött célunk tehát egy olyan Internetes hálózsem (Web-kiszolgáló / Web-hely) létrehozása, amely:

- Folyamatosan, jól és biztonságosan működik (nem törhetik fel kívülállók, és nem érhetik el / változtathatják meg fontos üzleti adatainkat)
- Teljességgel megfelel dinamikusan változó igényeinknek
- Teljes értékű és költségkímélő
- Nem igényel a felhasználók részéről beavatkozást a rendszergazdai feladatokba
- Fejleszthető, korszerű szoftverekkel van ellátva, melyek támogatása / fejlesztése nem fog megszűnni
- Gyorsan megvalósítható és könnyen karbantartható
- Már bizonyított, stabil rendszer, mely széleskörűen használt
- Önmagát felügyeli és értesíti a rendszergazdát hiba esetén

- Található hozzá rendszergazda, vagy a rendszergazda a rendszert hamar megtanulhatja
- A TCO (*Total Cost of Ownership*, a termék teljes birtoklási ideje alatti költség) nem haladja meg egy kereskedelmi termék TCO-ját.

Hogy ezek a feltételek igazak-e teljességében, az természetesen vitatható, hiszen egy folyamatosan fejlesztés alatt lévő, megújuló rendszerről van szó. Mivel világszerte fejlesztik, főleg angol nyelvű a rendszer, bár vannak már honosított részei. Természetesen a felhasználó (a Web-hely látogatója) számára ezek a dolgok láthatatlanok maradnak.

Ha megfelelően biztonságos és „önműködő” rendszert építünk ki, akkor a TCO is lecsökkenhet: „a gép forog, az alkotó pihen” (Madách), nem kell a rendszerhez hozzányúlni, ezért tényleges költség nincsen, csupán áramot kapjon a gép.

Bevezetés gyakorlati szempontból

Akik egy kicsit is járatosak az informatikában - és akiknek ez a munka jobbra szól – biztosan hallottak már a Linux-ról. Nekik nem nagyon kell elmagyaráznom, hogy ez mit fed, de azért sokakban keverednek a fogalmak. A Linux maga „csak” egy rendszermag (kernel), melyre egy főként GNU programokat használó operációs rendszer épül. Mivel a Linux POSIX⁸ kompatibilis (vagyis egy szabványos UN*X⁹ klón) ezért sok kereskedelmi UN*X-on (ezek hálózati operációs rendszerek) futó kereskedelmi szoftver újrafordítás után futtatható (portolható) reá. Napjainkban a Linux/GNU (továbbiakban Linux) páros már bizonyított: 1998-ról 1999-re 212%-os növekedést értek el a kereskedelmi Linux disztribúciók (terjesztések) eladásában. Ebben az adatban persze nincs benne a nagyobbik rész, hiszen ezeknek a rendszereknek egy része ingyenesen letölthető az Internetről, továbbá vannak nem kereskedelmi, Linux kernelre alapuló disztribúciók, pl. a Debian GNU/Linux, melyet a Debian projekt keretében fejlesztenek több mint ötszázan (csak a disztribúciót, a különböző szoftvereket több ezren.) világszerte szabadidejükben, ingyen és önkéntesen.

Kérdés az, hogy ezek az „otthon barkácsolt” szoftverek:

- Megfelelőek, stabilak, biztonságosak-e?
- Mennyire korszerűek, naprakészek, újszerűek?
- Mennyire vannak kitéve a vírusoknak?
- Mennyire könnyen feltörhetők ezek a rendszerek?

⁸ A POSIX a nemzetközi UNIX szabványosítási hivatal, ők határozzák meg melyik operációs rendszert lehet Unix-nak nevezni.

⁹ A UNIX az OpenGroup védjegye és azért, mert ezt itt mint fogalmat használom, szakmai szokás szerint UN*X-nak jelölöm, hogy ne keveredjen össze a konkrét termékkel, hanem a UN*X-okról, mint általános fogalomról legyen szó. Továbbiakban a UN*X szó a UN*X-ot és klónjait jelöli. (Részletesebben tájékozódjunk a <http://www.opengroup.org/trademarks.html> helyen.)

- Mennyire szabványosak, következetes-e a tervezésük?
- Mennyire kompatibilisek más szoftverekkel?

Bár sokan nem hiszik el, a Linux rendszerek nagy része (megfelelő rendszergazda mellett) sokkal biztonságosabb egyes kommerciális szoftvermegoldásoknál (többek között azért, mert a rendszergazdának teljes körű hatalma lehet a szoftver felett a forráskódon keresztül). A Debian disztribúció pedig régóta arról ismert, hogy az egyik legbiztonságosabb Linux terjesztés, kifejezetten az Internetre szánva. Ha egy cég az Internettől elzárt belső hálózatára vásárolna (pl. adatbázis-szervernek) egy gépet, akkor lehet, hogy jobb lenne egy kereskedelmi Linux terjesztést választania, mert a nagy üzleti adatbázis rendszerek érthető okokból ezeket jobban támogatják. A mi esetünkben azonban másra van szüksége kisvállalatunknak. Az Internet veszélyei miatt számunkra első a biztonság.

A Debian-nak mindig van egy stabil („*stable*”), lezárt változata, amelyet az éles rendszerekbe szánunk. Ebbe csak akkor jelennek meg programfrissítések, ha valami biztonsági hiba derül ki, de ekkor nagyon gyorsan. A Linux terjesztések közül általában a Debian-ban jelennek meg először a hibajavítások. Amint elegendő javítás összegyűlt, a stabil változatot újra kiadják. Pl. a „Slink” változat 6 kiadást ért meg. Az első az „r0”, az utolsó az „r5” volt. A másik változatot fejlesztőknek szánják („*unstable*”), melyek általában nem állják meg a helyüket éles munkában, bár vannak olyan türelmetlen rendszergazdák, akik mindig a legújabb programokat szeretik használni – helytelenül. Ebben a változatban szinte naponta frissülnek a programcsomagok és itt általában a különböző programok fejlesztői változatai találhatók. Ez lehetőséget ad a széleskörű tesztelésre és a gyorsabb hibajavításokra. Amint egy idő elteltével már eléggé stabillá válik a rendszer, akkor „kódfagyasztás” állapot („*frozen*”) következik be. Ekkor 1-2 hónapig „figyelik” a rendszert, és ha minden felmerült programhibát kiküszöböltek, akkor a változat „stabillá” válik, és CD-ROM-on is terjeszthető lesz. Ezután új fejlesztői változatba kezdenek a programozók. Mindegyik változat elérhető az Interneten FTP és több más protokoll segítségével az <ftp://ftp.debian.org-ról>, vagy a hivatalos (és az egyéb nem hivatalos) magyar ftp-tükörről: <ftp://ftp.hu.debian.org>. A programok futtatható és forráskód formában is letölthetők.

A Debian-t rendszergazdák és rendszerprogramozók fejlesztik – főleg rendszergazdák számára. Viszonylag nem olyan könnyen és gyorsan telepíthető, megtanulható azok számára, akik még nem ismerik a UN*X-os szemléletet. (E munka többek között ezért is íródott, hogy a kezdeti lépésekhez adjon segítséget.) Viszont cserébe teljesen személyre szabható és darabokra szedhető a rendszer.

A Linux / UN*X rendszerek általában teljesen immúnisak a vírusokkal szemben. Ez következik a rendszer többfelhasználós mivoltából és az alapoktól való viszonylag biztonságos tervezésből. Természetesen Linux alá is lehet kártevő programokat írni, de

ezek inkább a program férgek és a trójai falovak kategóriájába tartoznak.¹⁰ Az ilyen rendszerek elleni támadások nagy része inkább DoS (*Denial of Service* = szolgáltatás megbénítása) jellegű.

Sok szabad szoftver épp azért készült el az RFC (*Request for Comments*, Internet szabványokat leíró dokumentumok sorozata) leírások alapján, mert néhány üzleti jellegű szoftvercég ezeken (vagy ezek saját önkényesen módosított változatán) alapulva írta meg a szoftvereit. (Vagy esetleg később „de facto” szabványokká váltak bizonyos protokollok. Kitűnő példa erre a *SaMBa*, mely a *NetBIOS/SMB* hálózati protokoll UN*X-os szabad forráskódú implementációja, mely az adott RFC-k alapján készült). A szabad szoftverek (értsd: szabad forráskódú) legnagyobb részét éppen nem önkényesen, hanem minden szabványt és leírást, már működő programokat figyelembe véve kezdték el megtervezni és leködölni, hogy minél több más rendszerrel kompatibilis és használható legyen. Ezáltal egy Linux rendszer elég heterogén hálózati környezetben is kellően használható.

Rendszergazdai szempontból a kitűzött célunk tehát a következő:

- Megismerkedni a szükséges alapfogalmakkal, metódusokkal
- A döntéshozókkal együttműködve a felhasználási igény szerint megtervezni a rendszert
- Kiválasztani, megvásárolni és összeszerelni a hardvert, kiválasztani a megfelelő Internet-szolgáltatót, sávszélességet, előfizetni
- Beszerezni, telepíteni és konfigurálni a szoftvert
- Saját hardverhez és felhasználási igényhez optimalizálni a beállításokat
- A távoli karbantartás lehetőségének biztosítása
- A rendszer biztonságossá tétele a betörésekkel szemben
- A rendszer maximális rendelkezésre állásának biztosítása hardver és szoftver eszközökkel (biztonsági mentés, szünetmentes tápegység)
- A rendszer állapotának intelligens monitorozása, igénybevételi statisztikák készíttetése szoftverekkel

A gyakorlatiasabb fejezetekben feltételezem, hogy az olvasó a rendszergazdai szerepkört tölti be és ebből a szemszögből vizsgálom a problémákat.

A dolgozat első részében kisebb-nagyobb részletességgel kitérek az alapvető szabad-szoftver fogalmakra, többek között a GNU és más projektekre, azok filozófiájára,

¹⁰ [31. p. 28.] „Nagygépes rendszereknél [értsd: UNIX típusú] a védekezés egészen más filozófián alapul, mint PC-k esetében, hiszen itt nem csak tüneti kezeléssel (tehát féregirtással) vetnek gátat az újrafertőződésnek, hanem azzal is, hogy a biztonsági rendszeren található rést »befoltozzák«.” [31. p. 93.] „Erre a legjobb példa az alapvetően PC-re fejlesztett Linux melyet olyan komoly védelemmel láttak el, hogy nem érdemes rá vírust fejleszteni, mert az nem tudná igazán átvenni a vezérlést a rendszer feje felett.”

célkitűzéseire. Továbbá néhány hasznos és alapvető hálózat / Web specifikus struktúrára is, mint a fizikai elrendezés és a karbantartó személyzet felépítése.

A második fejezet a Tervezési fázissal foglalkozik: mit kell és mit érdemes létrehozni, mire van / lesz igény, mi a felhasználás területe, annak mérete, stb. Bemutatok egy mintapéldát, mely alapján a későbbi fejezetekben a megvalósítás történni fog. Továbbá foglalkozok a biztonság tervezésével is, hiszen ebben a szakaszban sok, később kiütköző probléma kiküszöbölhető. Megfelelően kidolgozott tervezési fázis után a megvalósításhoz és a teszteléshez sokkal kisebb idő kell, mint az *in medias res* típusú megvalósításoknál.

A megvalósítással foglalkozó fejezetben először a szoftver installálásának menetén vezetem végig az olvasót, majd az utólagos és lehetséges finomhangolásokról lesz szó. Tudni kell, hogy az installálás után a rendszer már teljességében működőképes, a finomhangolásra csupán a teljesítmény és a biztonság fokozásának érdekében van/lehet szükség.

A következő fejezetben a rendszer egy gyakorlatban már alkalmazott példáját vázolom röviden.

A fent tárgyalt rendszerkomponensek jövőjét is górcső alá veszem egy rövidebb fejezet erejéig, vajon a mostani és tervezett fejlesztések mit ígérnek számunkra.

Végül a Debian GNU/Linux rendszerben található egyéb, az adott célra alternatívát nyújtó szoftverek rövid bemutatását célozom meg.

II. Alapfogalmak

Ahhoz, hogy munkához tudjunk látni, szükségünk van néhány UN*X-os és Linux-os és operációs rendszerekhez kapcsolódó alapfogalomra. Ehhez ad segítséget *Kósa Attila* írása¹¹ is. Mivel ez a kérdés is igen terjedelmes kifejtést követelne, ezért elhagyom és inkább a téma fővonalára összpontosítok, feltételezve, hogy az olvasó már rendelkezik ezekkel az alapfogalmakkal. Kezdőknek ajánlom a [14] [15] [30] könyveket, és a *sysadmin-guide* csomagot¹², mely az LDP¹³ részeként írt Kezdő Rendszeradminisztrátorok Kézikönyve. A TCP/IP működésével kapcsolatban olvassuk el *Scheidler Balázs* írását.¹⁴ Ajánlom továbbá a függelékben szereplő szómagyarázatok áttekintését.

Fontos kiemelnem, hogy legtöbb hiba és félreértés abból ered, hogy a kezdő rendszergazda más, nem UN*X alapú rendszerekhez szokva nem tudja, hogy itt a sorrend a következő:

1. Olvasd el a dokumentációt (`/usr/doc` könyvtár alatt lévő fájlok, manuál oldalak)
2. Ha még ezután sem érted, akkor olvasd el a FAQ-kat¹⁵
3. Ha ez se segít, látogasd meg az adott program Web-helyét és kutass új dokumentációkért.
4. Ha már végképp nem értesz valamit, akkor kérdezz meg profikat a helyi Linux-os, és / vagy az adott programhoz tartozó saját levelezőlistákon.
5. Csak miután a témát már megértetted, akkor kezdj neki a program beállításának / éles használatának.

A legtöbb hiba abból ered, hogy az emberek feltelepítik a programokat és aztán azt hiszik, hogy eddigi nem UN*X-os tapasztalataik alapján tudni is fogják kezelni. Ezután pánikba esnek, hiszen valami nem úgy megy, ahogy kellene, a rendszer pedig már élesbe van állítva. Ekkor gyorsan írnak egy haragos hangvételű levelet egy listára, hogy ez meg az nem megy. Persze általában – a dokumentáció ismeretében – a megoldás triviális. (Ezért sokszor válaszképp, csak annyit kap, hogy „RTFM”¹⁶.)

Summa summarum, a szerver operációs rendszereket szakembereknek készítették és nem átlagos felhasználóknak. A kezdő legyen türelmes. Pár napon belül úgylis rájön mennyi mindent nem tud még a rendszeréről.

¹¹ <http://www.vmg.sulinet.hu/vmg/home/szamtech/linux/node62.htm> A munka kereteibe nem férne egy teljes referencia elkészítése, ezért ezt a forrást gyakran fogom használni, mint hasznos információ-forrást. A szerző Kósa Attila.

¹² <http://www.iki.fi/viu/linux/sag/>, TeX, DVI, HTML formában is, továbbá benne van a Debian-ban is

¹³ Linux Documentation Project

¹⁴ <http://www.vmg.sulinet.hu/vmg/home/szamtech/tcpip/>

¹⁵ Frequently Asked Questions: Gyakran Feltett / Ismételt Kérdések (GYIK)

¹⁶ Finomabban szólva: Read The FAQs and Manuals, olvasd el a dokumentációt.

1. A GNU projekt, a GPL licenz

A GNU projekt (és a *Free Software Foundation*) szabad szoftvereket fejlesztő programozókat fog össze. Ezek az emberek általában nem főállásban, csupán „hobbiból” írják meg ezeket a programokat. Azért teszik ezt, mert megunták, hogy a kereskedelmi programokat készítő cégektől függjenek. A GNU a szabadság ideáját közvetíti az emberek felé. Bár sokan kritizálták őket, „[...] ezek a programozók saját programjaikat továbbra is szabadon közreadták, várták mások módosító javaslatait, esetleg programrészeit, ezek közül a jobbakat beépítették az új verziókba, és így tökéletesítették programjukat. Ez többnyire jobb minőségű szoftverekhez vezetett, mint a nagy cégek korlátozott programozói gárdáinak termékei, amelyek erősen üzleti megfontolások szerint készülnek.

A sok különálló elszánt programozót szeretne volna Richard M. Stallman összefogni az 1980-as évek első felében azzal, hogy megalapította a »Free Software Foundation«-t (FSF, Szabad Szoftver Alapítvány), és elindította a »GNU project«-et. Előbbinek elsődleges célja, hogy alapítványként adományokat fogadhat el, amelyekből gépparkot tarthat fenn és fizethet a programozóknak, utóbbi magát a programozási munkát hivatott koordinálni. A GNU project alapvető célja, hogy egy teljesen szabadterjesztésű programokból álló, UNIX-szerű rendszert hozzon össze.”¹⁷

A GNU projekt célja tehát egy szabad forráskódú és ingyenes UN*X klón kifejlesztése. Mindez az Internet széleskörű elterjedésével lehetővé vált, ui. nem kell már se a székház, se a számítógépek, mert mindenki otthonról, a világ minden tájáról dolgozhat és segítheti a GNU projektet. Sok ember bár hivatalosan nem tagja a szervezetnek, de GNU/GPL licenz alatt adja ki a programjait, ezzel is támogatva a szabad szoftver közösséget.

Hogy mi is az a UN*X? - kérdezheti az olvasó, hiszen egyre azt emlegetjük, hogy a Linux az egy UN*X klón. Nos, ez általában véve kereskedelmi hálózati operációs rendszert jelent és nagyon sok fajtája van. Mivel magának a UN*X-nak is igen terjedelmes történelme van, ezért ennek részletezését mellőzöm. Ezen a címen¹⁸ találhat az olvasó egy jó magyar nyelvű összefoglalót. A lényeg az, hogy annak idején a UN*X-ból nőtt ki az Internet. A UN*X-klónok a kezdetektől (és az Internet¹⁹ kezdetétől) fogva hálózati operációs rendszerek, tehát kifejezetten erre a célra és nem egyedülálló személyi számítógépekre lettek kifejlesztve. Ennélfogva, ezek többfelhasználós, multitaszkos rendszerek. Bár a legősibb változatoknál még szóba sem került a biztonság - hiszen nem is volt még kiber-bűnözés, a mai változatok nagy részét már a tervezés során a biztonságra hangolják.

¹⁷ <http://www.vmg.sulinet.hu/vmg/home/szamtech/linux/node38.htm>

¹⁸ <http://www.vmg.sulinet.hu/vmg/home/szamtech/linux/node27.htm>.

¹⁹ Az Internet eredetileg a hálózatok közötti hálózatot jelentette, mely nem világméretűnek indult.

Maga a GNU filozófiájának kifejtése is meghaladja e munka kereteit. Viszont ezen a címen²⁰ magyarul olvashatjuk Richard M. Stallman gondolatait és érveit, a GNU kiáltványt. A függelékben olvasható a GNU GPL v2 licenz magyar nyelvű változata. További információkat szerezhetünk a <http://www.gnu.org>, <http://www.fsf.org> címeken.

Az is felmerülhet az olvasóban, hogy ezek az otthon barkácsolt szoftverek mennyire megbízhatóak. Erre a kérdésre Linus Torvalds így válaszolt: „[...] valamikor régen megjelent egy tanulságos beszámoló egy számítógépes terhelési próbáról, amely abból állt, hogy véletlenszerű, rossz adatokat tápláltak rendszerekbe, és figyelték, mi történik velük, hány száll el közülük. Kiderült, hogy az ingyenes segédprogramok sokkal ellenállóbbak, mint a fejlesztők termékei. Hogy miért? Mert sokkal több ember dolgozott rajtuk: sokkal többen odafigyeltek arra, hogy ellenállóbbak legyenek.”[32]

A másik ellenérv az szokott lenni, hogy ami ingyen van, ahhoz nincs támogatás, nincs semmire garancia, nincs akkor segítség se. Egy GNU/Linux szoftvereket tanuló felhasználó megnyilatkozása: „Találtam magyar nyelvű levelezési listákat, ahol számomra teljesen meglepő módon, időt és fáradságot nem kímélve, segítenek a kezdő felhasználóknak. Nagyon szokatlan volt először ez a segítőkészség, és még azóta is számtalanszor tapasztalom azt a remek érzést, hogy milyen jó segítséget kapni és adni. És mindezt anélkül, hogy tudnának rólad bármit is. Szinte hihetetlen! Ez az önzetlen segítőkészség áthatja az egész Linux mozgalmat.”²¹

Gyakori ellenérv még a dokumentáció milyensége. Minden szoftverhez részletes dokumentáció tartozik, néhány programhoz a Web-helyüket (vagy egy részét) is mellékelik. Sok dokumentáció fellelhető SGML, HTML, PDF, PS, DVI, TXT, stb. formátumokban is, melyek közül sok (pl. PDF, PS) nagyon szépen, jó minőségben kinyomtatható és máris kész a papíralapú leírás. Természetesen vannak magyar fordítások is, de ezek inkább az ún. "manual page"-ek²² fordításai és a HOGYAN (HOWTO) fájlok esetében igaz. A magyar fordítás értelemszerűen mindig elmaradhat az angol változattól. Ezért mindig az angol változat az érvényes.

A fordítók a Web-helyek fordításának is nekiláttak, pl. www.debian.hu cím alatt a www.debian.org fordítását találhatjuk.

Összegezve, véleményem szerint a jövő a szabad szoftvereké, s közöttük legjobban is a GPL licenszű programoké. Igaz sok tekintetben még nem versenyezhetnek a kereskedelmi programokkal, (pl. kezdő-felhasználók támogatása). Sokan olyan dolgokat várnak el ezektől a programozóktól, ami nem az ő feladatuk. Ezeket a programokat nem komplett számítástechnikai analfabétáknak írják, de a nagymamára

²⁰ <http://www.vmg.sulinet.hu/vmg/home/szamtech/linux/node48.htm>.

²¹ <http://www.vmg.sulinet.hu/vmg/home/szamtech/linux/node4.htm>

²² Manual Page: Kéziköny

ugyan ki bízna szervertelepítést. Otthoni felhasználásra még nem olyan széles körben használható, mint egyes kereskedelmi termékek (abban az esetben, pl., ha kezdőkről van szó).

2. A Linux kernel

A Linux egy szabadon terjeszthető (GPL) UN*X-klón rendszermag. A következő processzorokon / architektúrákon fut jelenleg (valamilyen módon és nem minden altípuson): ARM, AS/400, AP+, Apollo, DEC Alpha, MIPS, CE, ELKS (8086, 80286), mk86, MC68000 (PalmPilot), NeXT, PowerPC, SH*, PA-RISC, SGI, SUN *sparc, VAX, x86, stb.²³ A Linux-ot a kézi zsebgépektől a nagy szekrénynyi (Mainframe) gépekig szinte mindenre átültették, persze mi itt kis hazánkban anyagi és piaci okokból nem férünk hozzá sok ilyen hardverhez, így ez csupán érdekesség lehet számunkra.

A Linux igazából csak egy rendszermag, mely megfelel a POSIX szabvány előírásainak. Sokan összekeverik a Linux-ot a reál épülő komplett terjesztésekkel, mint amilyen pl. a Debian is. A többi szoftver, mely a kernel segítségével futtatható akár lehet kereskedelmi licenszelésű is. Azt, hogy milyen részei vannak egy operációs rendszernek, hogy mi a rendszermag feladata, egy – ebbe a témában alaposan elmélyedő szakkönyv elolvasásával tudható meg. Ilyen pl. a [4]. Végeredményben, a könyvben én is sokszor fogok a terjesztésekre „Linux” jelzővel hivatkozni, egyszerűsítésképpen.

A Linux-ot *Linus Torvalds* kezdte el fejleszteni 1990-ben, egy operációs rendszerről szóló egyetemi évfolyamdolgozatával. Népszerűségét a GPL licenz biztosítja.

„Magát a Linux operációs rendszert a GNU General Public Licence védi, ugyanaz a szerzői jogi copyright, amit a Free Software Foundation fejlesztett ki és alkalmaz. Ez az engedély bárkinek lehetővé teszi, hogy terjessze vagy módosítsa a szoftvert (díjmentesen, haszon nélkül), amíg a módosításai és bővítései szintén szabadon terjeszthetők. A »szabad szoftver« kifejezés a teljes szabadságra, nemcsak a jogdíjmentességre vonatkozik.” [1. p. 7.] „A Linux legfontosabb aspektusa, hogy maguk a felhasználók fejlesztik és bővítik a maguk igényeik szerint” [1. p. 5, Matt Welsh]

A Debian GNU/Linux 2.2 (Potato) kiadása a Linux 2.2.x-es kernelsorozatával van ellátva. A Linux kernel verziószáma a következőképpen néz ki: az első szám a Version, a második a Patchlevel, a harmadik a Sublevel és a negyedik az Extraversion (ha van). Ha a második szám páros, akkor ez egy stabil kernel, ha páratlan, akkor fejlesztői kernellel állunk szemben. Éles rendszerben csak stabil kernelt

²³Bővebben: <http://www.linux.org/projects/ports.html>

szabad használni, jelenleg a 2.2.x-est. Ez most a Potato-ban 2.2.16²⁴. A fejlesztői (2.3.x) változat gyorsan változik, és sokszor okoz kavarodásokat, esetleg rendszerösszeomlást is. Mire ez a mű az olvasó kezébe kerül, lehet, hogy már a 2.4-es sorozat vélhetőleg stabil lesz és a Debian Woody nevű fejlesztői változata tartalmazni fogja. A 2.4.0-testX változatok még nem számítanak stabilnak.

Néhány technikai jellegű információ

„A Linux soha nem volt 16 bites, az első perctől fogva 32 bitesnek tervezte szülőapja - Linus Torvalds. (Szülőanyjaként az Internetet szokták emlegetni.) Valódi többfeladatos működésű, egyidejűleg több felhasználót kiszolgálni képes operációs rendszer. [...] A Linux (az egyéb PC-s Unix-okhoz hasonlóan) nem használja a BIOS-t, mert a BIOS rutinjai úgy vannak megírva, hogy csak azután adják vissza a vezérlést, miután az I/O művelet befejeződött.

A fájlrendszer maximális mérete 2 TB, a maximális fájlméret 2 GB (a hivatalos kernelek még nem tudnak ennél nagyobb fájlt kezelni, de már készült patch, ami 16 TB-os fájlméretet is tud kezelni).

16 processzort támogat Intel platformon. A Linux nem csak Intel platformon képes futni, van más architektúrára írt változata is (SPARC, 64 bites SPARC, Alpha, PowerPC, MIPS, stb.). Természetesen más architektúrán is támogatja több processzor használatát, és ki is használja az általuk nyújtott teljesítményt.

A rendszer minimális hardverigénye: 80386SX 16-os processzor, 1 MB RAM és floppymeghajtó - merevlemez nélkül! Természetesen ilyen konfiguráción éppen csak működik a rendszer.

Létezik Linux 3Com PalmPilot-ra is.

A nagy méretekre is egy példa: SGI 36 processzorral, 5 GB RAM-mal, és egy másik Sparc alapú Fujitsu AP 1000 nevű számítógép 128 processzorral.

Látható, hogy nagyon széles a skála, amin a Linux elfut, és nagyszerű teljesítményt produkál. Nincsen semmilyen korlátozás a felhasználók számát tekintve, tehát maximum a hardver határolhatja be a kiszolgált felhasználók számát. A hálózati protokollok közül támogatja például az IPv6-ot.

A feladatütemezője prioritásos ütemezést használ, így különböző prioritásokat rendelhetünk a futtatandó programjainkhoz. A kernelmodulok dinamikusan betölthetők és kivehetők a memóriából [...]

²⁴ Fel kell hívni a figyelmet, hogy a 2.2.16-os változatban kijavítottak egy komoly biztonsági hibát, melynek segítségével helyi felhasználó rendszergazdai jogköröket szerezhet. Ezért csak a 2.2.16 vagy frissebb kernel használata javasolt. A Potato-ban lévő 2.2.16-os kernel sok olyan foltot is tartalmaz, mely hivatalosan csak a 2.2.17-esben lesz benne.

A 2000. év problémája Linux alatt nem jelent gondot. Ugyanis, mint a legtöbb UNIX, a Linux is 1970. január 1-je óta számolja az időt másodpercekben. Ezt az értéket egy 4 bájtos, előjeles számban tárolja, ami csak 2038-ban fog túlcsordulni²⁵.

A 64 bites Merced processzorra való áttérésről azt nyilatkozta Linus Torvalds, utalva az Intel nyitására a Linux felé, hogy »[...] ma már nem hiszem, hogy a Merced komoly kérdés volna. «[32]»²⁶

Igazság szerint a Linux kernelről már könyveket írtak és ráadásul a fejlődésével egyre többet tud – egyre több információ szerezhető meg róla. A mélyebb érdeklődésűek figyelmébe ajánlom a `linux/Documentation` könyvtárat, mely a forráskódban található és a legfrissebb információkat tartalmazza a kernel különböző részeiről. Számunkra a kernel inkább csak addig érdekes, amíg beállítjuk, lefordítjuk és futtatjuk. Utána már akkor jó, ha „észre se vesszük”, hogy létezik.

Elérhetőség:

A Linux kernel forráskódja a *Linkek*-ben szereplő helyekről tölthető le. Pl.: <ftp://ftp.hu.kernel.org/pub/linux/kernel/v2.2/linux-2.2.17.tar.gz>

Ez egy igen nagy méretű (kb. 15 MB) fájl. Lefordításához többek között szükség van a `binutils`, `(tar, gzip)`, `egcs/gcc` csomagokra. Hogy ne kelljen minden újabb kernelverzió kiadásakor letölteni ekkora méretű fájlt, ezért elég csak a frissítést letöltenünk. Pl.: <ftp://ftp.hu.kernel.org/pub/linux/kernel/v2.2/patch-2.2.18.gz>

A linux kernel részeit és lefordítását bővebben később tárgyalom.

Nemzetközi változat

Az USA-ban még érvényben lévő exportszabályozások miatt a titkosító algoritmusokat tartalmazó kódok egy külön európai gépen találhatóak. Ha szükség van, pl. fájl-rendszer szintű titkosításra, akkor töltsük le a kernelverzióinkhoz megfelelő foltot („patch”-et²⁷). A törvényi szabályozások azonban 2000-ben megváltoztak, így nemsokára már a titkosítás is belekerül a kernel nagy csomagjába.

Linkek:

<http://kernelnotes.org> linux kernel információk, a hivatalos kernel honlap

<http://edge.kernelnotes.org> a fejlesztői kernel honlapja

<http://netfilter.kernelnotes.org> a csomagszűrő modulok honlapja

<ftp://ftp.hu.kernel.org> a linux kernel hivatalos hazai ftp tükre

<ftp://ftp.kerneli.org> a linux kernel nemzetközi változatának ftp szervere

²⁵ És ezt addigra ki fogják küszöbölni, úgy, hogy 4 bájt helyett pl. 32 bájton fogják ábrázolni az időt.

²⁶ <http://www.vmg.sulinet.hu/vmg/home/szamtech/linux/node5.htm>

²⁷ Ezt nem igazán szokták lefordítani magyarra. Olyan toldozás-foltozást jelent, mellyel egy nagyobb forrás könyvtárat egészíthetünk ki a nekünk megfelelő opcionális forráskódokkal.

3. A Debian projekt

A Debian projekt²⁸ célja egy olyan szabad szoftverekre épülő teljesen ingyenes terjesztés (vagy inkább kernelfüggetlen operációs rendszer) kifejlesztése, amely nem kereskedelmi célú és független irányító testülettel rendelkezik. Ez a rendszer nem tartalmaz semmilyen kizárólagosan kereskedelmi szoftvert. Jelenleg egy stabil (Debian GNU/Linux) és két fejlesztői változata (Debian GNU/Hurd, Debian GNU/FreeBSD²⁹) van a kernelek szempontjából.

Mit jelent ez a szó?

„Mivel sokan kérdezték: a »Debian«-t »debián«-nak kell ejteni. A név a Debian alkotójának, Ian Murdocknak és feleségének, Debrának a nevéből származik.”³⁰

„Mi az a Debian?”

A Debian szabad vagy nyílt forráskódú számítógépes operációs rendszer. Az operációs rendszer alapvető programok összessége, amelyek a számítógép működéséhez szükségesek. Az operációs rendszer magja a kernel. A kernel a számítógép sarkalatos programja, amely az alapfeladatokat végzi, és más programokat indít. A Debian kernelfüggetlen. Jelenleg a Linux-kernel-t használja, de készülöben van a Debian más kernelekhez is, például Hurd-ra.”³¹

A Debian GNU/Linux a következő rendszereken fut: Intel x86 („i386”), Alpha („alpha”), ARM („arm”), Motorola 68k („m68k”), MIPS, PA-RISC, Motorola/IBM PowerPC („powerpc”), Sun SPARC („sparc”), Sun UltraSPARC („sparc64”),

Mire jó egy terjesztés?

Az alapprobléma az, hogy a szabad szoftverek főleg forráskód formában hozzáférhetőek az Interneten. Ahhoz, hogy összeállítsuk a rendszerünket, le kellene fordítani az összes programot és felinstallálni a célgépre. Ez rengeteg munkát igényelne. Ezért találták ki a terjesztéseket. Ezek olyan lefordított és előrecsomagolt programokkal rendelkeznek, melyek már előre be vannak konfigurálva egy adott működőképes és általános felhasználói profilra. Ezeket a beállításokat természetesen meg kell változtatnunk a saját igényeink szerint.

Fontos továbbá az is, hogy a terjesztések rendelkeznek ún. installáló / telepítő programokkal, indítólemezekkel, stb. Ezek nélkül körülményes lenne a rendszer telepítése. Tehát a terjesztés (disztribúció) az egy keret a szabad szoftverek

²⁸ Egyéb Debian specifikus adatok a függelékben.

²⁹ Ez igen nagy eszmei vitát váltott ki a fejlesztők között, hiszen a Debian-t átültetni a FreeBSD kernelére elméletileg ütközik a GNU elveivel, hiszen az BSD licensz alá esik.

³⁰ www.debian.org/intro/about.hu.html

³¹ <http://www.debian.org/intro/about.hu.html>

tömegéhez, mely egy egységbe kovácsolja azokat, miután azok egysége már nyugodtan nevezhető operációs rendszernek.

Minden terjesztésnek kell, hogy legyen egy ún. csomagkezelő programja, mely az előre lefordított bináris programokat tömörített formában tartalmazó csomagokat menedzseli. A Debian és leszármazottainak csomagkezelője a `dpkg`, a Redhat³² és leszármazottainak programját pedig `rpm`-nek nevezik. A Debian-ban is telepíthetünk `.rpm` formátumú csomagokat, de ajánlatos az `alien` nevű programmal átkonvertálni `.deb` formátumra. (A Debian csomagok `.deb`, a Redhat-osok `.rpm` kiterjesztésűek) Véleményem szerint a Debian csomagkezelője sokkal fejlettebb és jobban kidolgozott rendszer, mint bármelyik másik. Néhány fontos dolog a csomagkezelőről:

„A `dpkg` a Debian GNU/Linux csomagjainak installálására, eltávolítására, építésére és menedzselésére alkalmas csomagkezelő. Kérhetünk információkat a csomagokról. Egy csomagnak több státusza lehet:

- *installed* - feltelepített és konfigurált csomag,
- *half-installed* - a csomag telepítése el lett kezdve, de valami miatt nincs tökéletesen feltelepítve,
- *not-installed* - a csomag nincs feltelepítve a rendszerre,
- *unpacked* - a csomag fel van telepítve, de nincs konfigurálva,
- *half-configured* - a csomag fel van telepítve, de nincs teljesen konfigurálva,
- *config-files* - a csomag már nincs a rendszeren, csak a konfigurációs fájlok vannak meg.

Egy csomag kiválasztásának három státusza lehet:

- *install* - kiválasztva telepítésre,
- *deinstall* - kiválasztva törlésre,
- *purge* - minden része kiválasztva törlésre, még a konfigurációs fájlok is.

(Ugyanis a „deinstall” nem távolítja el a csomaghoz tartozó konfigurációs fájlokat.)

Egy csomagnak két jelzője lehet:

- *hold* - nem változtat a csomagkezelő az így megjelölt csomag állapotán,
- *reinst-required* - a csomag meg van sérülve, de nincs eltávolítva, ezért szükséges újratelepíteni.”³³

További fontos információk:

„Hol találok Debian-os infókat, csomaglistát, ilyesmiket?

A Debian website a <http://www.debian.org> címen található, de érdemes a legfelső sorban³⁴ látható tükrözések közül a legközelebbit választani (ez az Internet kapcsolatotól függ). A

³² A Redhat egy másik, kereskedelmi Linux terjesztés.

³³ <http://www.vmg.sulinet.hu/vmg/home/szamtech/linux/node231.htm>

csomagok között a <http://www.debian.org/packages.html> címen lehet keresgélni, akár a csomag nevét a felső ablakban (a teljes nevet meg kell adni!) vagy a csomag leírásában lehet az alsóban keresni. Az aktuális hibalista csomagokra, sürgősségre vagy más szempontok szerint lebontva a <http://www.debian.org/Bugs/> címen érhető el. Az angol nyelvű Debian FAQ a <http://www.debian.org/cgi-bin/fom> címen érhető el.

A Debian-os levelezőlistákban a <http://www.debian.org/Lists-Archives/> címen lehet keresgélni. A teljes disztribúció az alábbi helyeken érhető el: [...] <ftp://ftp.kfki.hu/pub/linux/debian/>³⁵, illetve ha valami probléma van akkor természetesen az <ftp.debian.org> címen.³⁷

A Debian GNU/Linux különböző kiadásait verziószámokkal és kódnevekkel jelölik. Itt a magyar Debian FAQ-ból idézek: „Mik ezek a furcsa nevek: bo? hamm? ...?

Ezek a kódnevei a különböző Debian verzióknak, amíg dolgoznak rajtuk. Kiadáskor mindegyik kap egy verziószámot, de a kódnevek általában az igazi rajongók között tovább élnek (a könyvtárnevekről nem is beszélve).

A jelenlegi verziók a következők: `buzz v1.1 rex v1.2 bo v1.3 hamm v2.0 slink v2.1 potato v2.2 woody v2.3` és `sid` ez az új architektúrák (`sparc`, `arm`, ...) gyűjtőhelye

Bruce Perens, a Debian ex-vezéralakja a Pixarnál (<http://www.pixar.com/>) dolgozik, akik a Toy Story című számítógéppel készült animációs filmet készítették. A jelenlegi kódnevek a Toy Story szereplői...³⁸

A Debian csomagkezelője kiegészül az `apt` nevű programmal, mely a folyamatos frissítést és a csomagok letöltését teszi lehetővé pl. az interneten lévő ftp helyekről.

Egy Debian tükör a következőképpen néz ki: a `/debian` könyvtár alatt találhatóak a Debian-nal kapcsolatos anyagok. Az ez alatt lévő `dists` könyvtár tartalmazza a Debian terjesztés különböző változatait. Pl. a `potato` alkönyvtár alatt a Potato változat csomagjai vannak. Itt három részre szakad licensz szerint: `main` (minden amit a DFSG³⁹ szerint szabadnak minősül), a `contrib` („olyan ingyenes csomagok, amik függhetnek nem ingyenes csomagoktól”⁴⁰) és a `non-free` („valami miatt nem ingyenes”⁴¹ alkalmazások, általában egyéni felhasználók számára ezek is ingyenesek”). Platform szerint megosztva vannak a bináris, és egy külön könyvtárban a forrás csomagok. Pl. `binary-i386` alatt vannak az Intel processzorokra szánt változatok. Ezek alatt alszekciók találhatóak. Pl. a `mail` könyvtárban a levelezéssel kapcsolatos programcsomagok vannak elhelyezve.

Tehát: <ftp.hu.debian.org/debian/dists/potato/main/binary-i386/mail>

³⁴ Értsd: az adott Web-oldal legfelső sorában.

³⁵ Én inkább az <ftp://ftp.hu.debian.org/debian> helyet ajánlom.

³⁷ Debian FAQ magyarul <http://mlf.linux.rulez.org/cgi-bin/fom?file=126>

³⁸ <http://mlf.linux.rulez.org/cgi-bin/fom?file=127>

³⁹ Debian Free Software Guide, avagy mit tart a Debian szabadnak. Részletesen a függelékben.

⁴⁰ <http://www.vmg.sulinet.hu/vmg/home/szamtech/linux/node230.htm>

⁴¹ Vagy esetleg nem nyílt forráskódú

Ezzel nekünk általában nem kell törődnünk különösebben, az `apt`⁴² program megtalálja egy `Packages.gz` nevű fájllista szerint, hogy milyen csomagok találhatóak az adott ftp tükrön. Ebből eldöntheti, hogy van-e csomagfrissítés a mi gépünkön lévő állapothoz képest. Ez a lista pontosan tartalmazza a programcsomagok relatív elhelyezkedését.

Nagyon fontos a Debian terjesztésben a `dselect` program. Ez egy menüs csomagtelepítő és karbantartó szoftver. Először megkérdezi azt, hogy milyen forrásból kívánjuk a csomaglistát és a csomagokat megszerezni. Ez lehet ftp, kompakt lemez, NFS⁴³ és a kedvelt `apt` program is. Én mindenképp az `apt` használatát javaslom. Persze ha van kompakt lemezünk és az kellően friss, akkor azt is alkalmazhatjuk. (Általában a kompakt lemezeket telepítéskor használjuk, frissítésnél pedig az ftp tükröt, bár lehet telepíteni közvetlenül az ftp-ről is.)

Ha ez megtörtént, akkor megszerzi a csomaglistát. Ennek birtokában már nekikezdhetünk válogatni. Bizony, ha kezdő Linux-os felhasználók vagyunk, akkor fájhat most a fejünk, ugyanis majd 4500 programcsomag közül kell kiválasztanunk a számunkra szükségeseket. Ez akár órákig is eltarthat, ha minden csomag rövid leírását át akarjuk tanulmányozni.

4. Az Apache projekt, az Apache moduljai

Az Apache projekt (<http://www.apache.org>) célja egy olyan Web-szerver program létrehozása és karbantartása, fejlesztése, amely megfelel a gyorsan változó Internetnek, elég biztonságos és üzleti célra is megfelelő és szabadon használható. Az Apache-ot, mivel a régi NCSA⁴⁴ `httpd` szerveréből készült, BSD licenz alatt terjesztik⁴⁵. A projektet rendszergazdák kezdték el amikor *Rob McCool*, az NCSA szerverének írója kilépett az NCSA-tól, és a szoftver nem fejlődött tovább.

Az Apache projekt koordinálását az *Apache Group* végzi. Néhány vezető és több száz fejlesztő van e mögött a szoftver mögött. Jelenleg a stabil változata 1.3.12, a fejlesztői pedig a 2.0a számot viseli. A Potato-ban jelenleg az 1.3.9-es verzió van és valószínűleg ez is marad meg, mivel ez már egy bevált, tesztelt verzió. A Woody változatban valamelyik újabb verzió lesz. Hacsak nincs valami különösebb okunk rá (pl. hibajavítás, teljesítménygondok) ne fordítsunk újabbat.⁴⁶

Az Apache Web-szerver szinte minden UN*X platformra lefordítható, futtatható, de egyéb kommerciális platformokon is működik .

⁴² Acquisition Package Transfer

⁴³ Network File System, a UN*X klónok hálózati fájlrendszere

⁴⁴ NCSA: National Center for Supercomputing Applications, University of Illinois,

⁴⁵ A függelékben olvasható ennek a licensznek a magyar nyelvű változata.

⁴⁶ <http://www.cert.org/advisories/CA-2000-02.html> érdemes áttanulmányozni ezt a dokumentumot. Ha ilyen tartalmú helyet üzemeltetünk, ne ijedjünk meg, mert a Debian fejlesztői a probléma megoldását „backport”-olták az 1.3.9-es verziójú csomagba. <http://www.apache.org/info/css-security> : ez az oldal is fontos információkat közöl a problémáról.

Az SSL bővítést *Ben Laurie* készítette az *OpenSSL* programkönyvtár segítségével. Érdeemes végigtanulmányozni a `/usr/doc/apache-common` és a `/usr/doc/apache-ssl` könyvtárakban lévő dokumentációkat és szerzői jogi információkat, példafájlokat.

Néhány alapvető fogalom:

Virtualhost: Ugyanazon a gépen több virtuális Web-szerver is futhat. Pl. a www.borgyar.hu és a www.szormegyar.hu ugyanazon a szerveren van, de kintről két különböző helynek látszik. Nagyon egyszerű egy gépen akár több száz virtuális szervert futtatni. Csupán a konfigurációs állományokat kell megfelelően behangolni. Minden egyes szerver teljesen különböző formában is beállítható, azok egymástól elég különbözően is viselkedhetnek. A virtuális Web-szerverek lehetnek külön-külön IP címen (*IPVirtualHosting*), vagy egy azonos IP címen is (*NameVirtualHosting*) egy azonos gépen. Tehát elég akár csak 1 db statikus IP címet vásárolnunk és több domént bejegyeztetni. Ha minden doménnek külön IP címet veszünk, nagy forgalom esetén később szétválaszthatjuk külön gépre is a helyeket.

SSL: Secure Socket Layer, vagyis Biztonságos Csatorna Réteg. Ennek a segítségével a kliens böngészője és a Web-szerver között titkosított formában fog folyni az adatcsere, ha mindkettő képes erre és úgy van beállítva. Ez nagyon hasznos érzékeny adatok cseréjekor, hiszen egyébként minden vonal nagyon könnyen lehallgatható. Főleg személyes és hitelkártya adatok cseréjekor szokták alkalmazni. Én javaslok ennek minél szélesebb körű alkalmazását a személyiségi jogok védelmében, hiszen az ember könnyen kiismerhető arról, hogy milyen tartalmakat látogat.

A titkosításnak a böngésző szempontjából 56-bites és 128-bites kóderősségű változata van. (Az USA exportszabályozásai miatt). A szerver oldalon szükség van egy úgynevezett igazolásra (*Certificate*) is, mellyel a szerver igazolja magát a kliens felé, hogy ő kicsoda-micsoda. Egy ilyen igazolást készíthetünk magunknak is, de ezt a böngésző gyanúsán fogja fogadni. Az igazolást általában egy kereskedelmi cégtől lehet vásárolni, mely egy adott időre szól. Ez a cég igazolja, hogy az igazolásunk valós adatokat tükröz és nem egy *cracker* banda *DNS spoof*-olt⁴⁷ ál-szerverére lépett be a felhasználó. Persze ezek a „*Signer*” cégek főként külföldiek és viszonylag sokba is kerül egy ilyen igazolás, ezért, ha szükséges, készíthetünk magunknak egyet. (Részletesen később.)

user authentication: vagyis a felhasználó azonosítása. Ha vannak olyan oldalak, ahová csak bizonyos felhasználók léphetnek be, akkor oda általában valamilyen

⁴⁷ Lásd: II. Alapfogalmak / 8. Biztonsági alapok (hardver, szoftver)

autentikáció szükséges. Ilyen lehet pl. egy név / jelszó bekérés, de ettől sokkal komplexebb formák is vannak. Azonosítás történhet LDAP⁴⁸ protokoll vagy adatbázis segítségével is.

Modulok: Nagyon hasznos funkció, hogy csak azokat a részeket kérjük betölteni a szerverprogram indulásakor (a konfigurációs fájl szerint), melyek számunkra szükségesek. Így rengeteg erőforrást takaríthatunk meg. Mivel a rendszer szabványos API⁴⁹-val rendelkezik, külső beszállítóktól is szerezhethetünk be speciális modulokat (akár kereskedelmi termékeket is), melyeket a rendszerbe beilleszthetünk. *“A kernelhez hasonlóan, az Apache is képes megfelelően elkészített külső file-t beültetni, mely a saját kódjába épül a memóriában. Az itteni modulok annyiban eltérnek a kernelben használatosaktól, hogy csak induláskor tölthetők be, futás közben nem.”* [19. p. 87] Ekkor persze a gépet nem kell újraindítani, csupán a Web-szerver programot.

Általában elég összetett kérdés, hogy melyik modul hol van, mi a funkciója, és hogyan illeszthető bele a rendszerbe. Mindenesetre a főág moduljai megtalálhatóak a <http://modules.apache.org> –on.

Pl. a `mod_rewrite` modul az oldalak kódjában lévő URL-ek átírásával foglalkozik. Ez bár elsőre viszonylag bonyolult dolognak⁵⁰ tűnik, megfelelő dokumentációval rendelkezik. Ez a probléma főleg a webgazda (lásd később) feladatába tartozik. A lényeg az, hogy ha megváltoztatjuk a Web hely „fizikai” felépítését, az oldalakban szereplő hivatkozásokat is mind meg kellene változtatni, ami igen nagy munka lehet. Ezért találták ki ezt a kitűnő segédeszközt.

A modulok⁵¹ listáját és funkcióját mutatja az 1. táblázat - Az Apache moduljai

<code>mod_access</code>	Gép (IP) alapú hozzáférés-szabályozás
<code>mod_actions</code>	Fájltípus / metódus alapú script indítás
<code>mod_alias</code>	Álnevek és átirányítások
<code>mod_asis</code>	Az „asis” fájl kezelő
<code>mod_auth</code>	Felhasználó azonosítás szöveg fájlokkal
<code>mod_auth_anon</code>	ftp stílusú névtelen felhasználó azonosítás
<code>mod_auth_db</code>	felhasználó azonosítás a Berkeley-féle DB (adatbázis) fájlokkal
<code>mod_auth_dbm</code>	felhasználó azonosítás DBM (adatbázis) fájlokkal
<code>mod_auth_digest</code>	MD5 felhasználó azonosítás
<code>mod_autoindex</code>	Automatikus könyvtár listázás

⁴⁸ Lightweight Directory Access Protocol, címtárkezelés

⁴⁹ Application Programming Interface, alkalmazás-programozási felület

⁵⁰ <http://www.apache.org/docs/misc/rewriteguide.html>

⁵¹ A táblázat nem teljes, mivel a dokumentáció egy lépéssel le van maradva a kódtól. Több olyan modult is találtam, amely nincs benne az általános dokumentációban.

<i>mod_cern_meta</i>	HTTP fejléc metafájlok támogatása
<i>mod_cgi</i>	CGI ⁵² script-ek meghívása
<i>mod_digest</i>	MD5 felhasználó azonosítás
<i>mod_dir</i>	Alapszintű könyvtárkezelés
<i>mod_env</i>	Környezeti változók átadása CGI script-eknek
<i>mod_example</i>	API demonstráció
<i>mod_expires</i>	Fejlécek erőforrásokhoz (meddig érvényes egy oldal)
<i>mod_headers</i>	Tetszőleges HTTP fejlécek beépítése
<i>mod_imap</i>	Kép-térkép fájlkezelő
<i>mod_include</i>	Szerveroldalon előállított objektumok
<i>mod_info</i>	Szerver beállítási információk
<i>mod_log_agent</i>	A „User Agent”-ek naplózása (a felhasználó milyen böngészőt használ)
<i>mod_log_config</i>	Testre szabható naplózó
<i>mod_log_referer</i>	„referer” naplózó (honnan lett a kliens erre az oldalra irányítva)
<i>mod_mime</i>	Objektumtípus megállapítása fájlkiterjesztésből
<i>mod_mime_magic</i>	Objektumtípus megállapítása tartalomról
<i>mod_mmap_static</i>	Fájlok térképének elkészítése a memóriába a gyorsabb kiszolgálás érdekében
<i>mod_negotiation</i>	Tartalom „tárgyalás”
<i>mod_proxy</i>	HTTP gyorsítótár
<i>mod_rewrite</i>	Profi URL-ből fájlnévre konvertáló
<i>mod_setenvif</i>	Környezeti változók beállítása kliens információk alapján
<i>mod_so</i>	Futás közbeni modul-betöltés támogatása (fejlesztés alatt!)
<i>mod_speling</i>	Automatikus hibajavítás félregépet URL-ekben
<i>mod_status</i>	a szerver állapotának megjelenítése Web-lapként (autentikáció szükséges)
<i>mod_userdir</i>	A felhasználók könyvtárait kezeli (listázás, letöltés)
<i>mod_unique_id</i>	egységes kérés azonosító generálása minden lekéréshez
<i>mod_usertrack</i>	Felhasználó-követés sütik (cookies) segítségével
<i>mod_vhost_alias</i>	dinamikusan konfigurálható virtuális szerver-támogatás

1. táblázat - Az Apache moduljai

Ha újra szeretnénk fordítani az Apache-ot (pl. hogy a rendszerünkhöz optimalizáljuk), ajánlom, hogy a Debian előrecsomagolt forrását használjuk, mert ekkor egyből, a Debian segédeszközeivel `.deb` csomagba fordíthatjuk azt. Ez nagyban megkönnyíti a későbbi használatot.

⁵² Common Gateway Interface, Általános Átjáró (kapcsolatteremtő) (programozási) Felület

Hogy miért pont az Apache-ot választottam?

Azért, mert:

- a világ Internetes Web-szervereinek 60%-án ez fut⁵³
- széles körben elterjedt, ismert, bevált, tesztelt
- ezt ismerik azok, akiktől segítséget kérhetünk (pl. levelező listák)
- nagy teljesítményű, rengeteg funkcióval és lehetőséggel rendelkezik
- ezután is támogatott, fejlesztett lesz még sokáig
- ingyenes, szabadon felhasználható, nyitott forráskódú
- elég biztonságos, ha jól be van állítva
- nagyon sok platformon fut, ezért egységesen használható
- nagymértékben személyre szabható
- nagy, neves gyártók által is támogatott

5. A Web-szerver helye a hálózatban (Internet/intranet)

A publikus Web-szerverünket az un. demilitarizált, semleges zónában kell elhelyeznünk. Ezt az 1. kép - A Web szerver helye a hálózatban mutatja.

1. kép - A Web szerver helye a hálózatban

„Egy tipikus profit szférába tartozó hálózat védekezésénél szigorúan konfigurált tűzfal rendszereket alkalmaznak, amelyek vagy gondosan kontrollálják-monitorozzák a kifelé irányuló hálózati forgalmat, vagy gyakran teljesen megakadályozzák azt. Az Internet felé nyújtott publikus szolgáltatásaikat olyan rendszerekről nyújtják, amelyek egy tűzfal mögött, de még a második tűzfallal védett belső hálózatukon kívül helyezkednek el (semleges zóna). Ha lehetővé tesznek interaktív belépést, akkor azt csak bizonyos hálózatokról és/vagy csak szigorú ellenőrzés után engedik meg.”⁵⁴

⁵³ www.netcraft.com/survey, 2000. márciusi eredmények

⁵⁴ Kadlecsek József: Védekezési szintek akadémiai hálózaton

kadlec@sunserv.kfki.hu, <http://www.iif.hu/rendezvenyek/networkshop/98/eloadas/html/h/jkadlec/jkadlec.htm>

A lényeg az, hogy az Internet felé nyújtott publikus szolgáltatásokat sose keverjük össze a csak belső használatra kialakított szolgáltatásokkal, azok ne legyenek közös gépen és alhálózaton.

Fontos beállítani a Web-szerveren is a csomagszűrést (pl. *ipchains*⁵⁵), hogy ez ne legyen egy ugródeszka a belső hálózat felé. Az Internet és a Web-szerver közötti szűrőn (lehet ez egy Linux-os tűzfal is PC-n.) tiltsunk le először minden forgalmat, amely befele irányul, majd engedélyezzük a következőket a Web-szerver felé (*--destination* IP címmel megadva!):

port	szolgáltatás	irány	miért?
80	http (Web)	kintről	ezt szolgáltatjuk
443	https (Web)	befelé	ezt szolgáltatjuk
22	ssh		távoli menedzsment
25	smtp (mail)		kapjunk leveleket

2. táblázat - Az engedélyezett szolgáltatások listája

A Web-szerverünkön más szolgáltatást az Internet felé nem kívánunk nyújtani. Az SMTP-re is csak azért van szükségünk, hogy a rendszergazda megkapja a rendszernaplókat és esetleg a rendszergazdák egymás között levelezhessenek.

Figyelnünk kell azonban arra is, hogy így a Web-szerverről nem fogunk tudni elérni semmit (pl. nem tudunk *ftp*-zni, *ssh*-zni), mivel a kapcsolódási portok (is) le vannak tiltva. Ezért hozzá kell még adnunk néhány olyan szabályt is, amely ezt lehetővé teszi.

Meg kell akadályoznunk továbbá azt, hogy olyan csomagok átjussanak a szűrőn, mely külső helyről érkező csomag fejlécében, a forrás (*source*) IP cím a mi belső hálózati szegmensünk valamely címét tartalmazza. Vagyis belső címről érkező csomagnak tünteti fel magát, de olyan interfészről érkezik, amely biztosan a külső hálózathoz tartozik.

Ha Linux-os tűzfalunk van így kell eljárni:⁵⁶ (Az *ipchains* program részletes bemutatására nem térek ki. Ennek használatát olvassuk el a dokumentációból. A HOWTO⁵⁷ 7. fejezetében részletesen tárgyalva van egy – a mi példánkhoz hasonló – hálózat tűzfalának beállítása. Az *ipchains*-el kapcsolatban ajánlom továbbá ezt a két cikket [22], [23].)

Először is, állítsuk be az alapviselkedést tiltóra:

```
ipchains -P input DENY      # minden olyan csomag amely nem felel meg egyetlen
ipchains -P output DENY    # további szabálynak sem, ne legyen átengedve
ipchains -P forward DENY   # se be, se ki, se továbbítva
```

Minden olyan csomagot, amely a *loopback* interfésznek fenntartott címről jön, de más interfészről, tiltsunk le, mert ez IP átejtés (*spoofing*).⁵⁸ Több hálózati kártya esetén

⁵⁵ Az *ipchains* program a Linux kernel csomagszűrését állítja be. Részletesebben később.

⁵⁶ Először olvassuk el az *ipchains*-HOWTO-t: `/usr/doc/netbase/ipchains-HOWTO.txt.gz` ! Majd a Firewall-HOWTO-t. (Lásd később.)

⁵⁷ Az *ipchains* HOWTO magyar fordítása: <http://www.rkcs.hu/linux/index2.html> Szabó Dániel owo@zed.hu

⁵⁸ A Debian Potato-ban ezt automatikusan megteszi a rendszer, ha úgy van beállítva!

tegyük meg ezeket az egyes interfészekkel is, vagyis pl. az Internethez tartozó interfészen ne jöhessen be olyan csomag, melynek forráscíme a belső hálózatunkhoz tartozik.

```
ipchains -A input -j DENY -l -s 127.0.0.0/8 -i ! lo
# a $MYNET/MASK helyett mindenki írja be a saját alhálózatát és annak maszkját.
ipchains -A input -j DENY -l -s $MYNET/MASK -i ! eth0 # stb.
```

Itt megengedjük, hogy az Internetről látható legyen a Web-szerverünk:

```
# az $INETIF helyett mindenki írja be azt a hálózati interfészt, mely az Internetre
# kapcsolódik
ipchains -A input -p all -i $INETIF -j ACCEPT -d „a mi Web-szerverünk IP címe” 80
```

Ha egy adott IP címről vagy tartományról nem akarjuk fogadni a kéréseket, akkor azokat tiltsuk le. (Pl. innen betörési kísérletek voltak már.)

```
ipchains -A input -p all -j DENY -s „akármi” -d „a mi Web-szerverünk IP címe” 80
```

Minden egyéb próbálkozást tiltsunk le, hogy senki se futtathasson a belső hálón Web-szervert úgy, hogy azt kívülről látni lehessen.

```
ipchains -A input -p all -j DENY -s 0.0.0.0/0 -d $MYNET/MASK 80
```

Ugyanezt tegyük meg a 443-as porton is.

```
ipchains -A input -p all -i $INETIF -j ACCEPT -d „mi Web-szerverünk IP címe” 443
ipchains -A input -p all -j DENY -s „akármi” -d „a mi Web-szerverünk IP címe” 443
ipchains -A input -p all -j DENY -s 0.0.0.0/0 -d $MYNET/MASK 443
```

Hasonlóképpen járhatunk el a 22-es és a 25-ös portok esetén is. Ha van másik levelező, vagy menedzselni való szerverünk, akkor adjunk meg megengedő szabályokat azokra is. Újra felhívom a figyelmet arra, hogy ekkor csak a Web-szerver 22, 25, 80, 443-as portjai érhetőek el, ezért egyrészt nem lehet a Web-szerverről kapcsolódni más gépekre és a DMZ-ben lévő más gépekre se, ezért azokat az IP-ket és portokat külön engedélyezni kell. Ha azt akarjuk, hogy vissza is kapjunk adatokat, ha belülről kérünk kifelé, tegyük ezt:

```
ipchains -A input -p tcp -j REJECT -y -i $INETIF -d „saját alháló v. gépcím”
ipchains -A input -p tcp -j ACCEPT -i $INETIF -d „saját alháló v. gépcím”
```

Ekkor a kívülről jövő külön nem felsorolt kapcsolatkéresek el lesznek utasítva egy „célállomás nem elérhető” válasszal. Amint látszik, ez csak a TCP protokollt igénylő szolgáltatások esetében igaz. Az UDP-s szolgáltatásoknál nincs engedély. Meg kell engednünk viszont az Internet és a belső szegmensek között az 53,123-as UDP portokat, mert ezek nélkül az egyes szolgáltatások nem működhetnek helyesen.

```
ipchains -A input -p udp -j ACCEPT -i $INETIF -d „saját alháló v. gépcím” 53
ipchains -A input -p udp -j ACCEPT -i $INETIF -d „saját alháló v. gépcím” 123
```

Az itt felsorolt szabályok csak töredékei az összes tűzfalon beállítandó szabályoknak. Célszerűbb a különböző zónáknak saját szabályláncot létrehozni. Erről bővebben a dokumentációban.

Figyelnünk kell továbbá az ICMP forgalmat is. Javasolt az „*echo-reply, echo-request, time-exceeded, destination-unreachable, source-quench*” típusú csomagok átengedése az összes többinek pedig a tiltása a tűzfalon.

```
ipchains -A input -p icmp -i $INETIF -j ACCEPT -d $MYNET/MASK --icmp-type echo-reply
ipchains -A input -p icmp -i $INETIF -j ACCEPT -d $MYNET/MASK --icmp-type echo-request
ipchains -A input -p icmp -i $INETIF -j ACCEPT -d $MYNET/MASK --icmp-type time-
exceeded
ipchains -A input -p icmp -i $INETIF -j ACCEPT -d $MYNET/MASK --icmp-type destination-
unreachable
ipchains -A input -p icmp -i $INETIF -j ACCEPT -d $MYNET/MASK --icmp-type source-
quench
```

Fontos továbbá az is, hogy a DMZ és a belső hálózat közötti tűzfalon csak azokat a portokat engedélyezzük átmenni, amelyek a cég munkájához szükségesek lehetnek. *„Csak azokat a szolgáltatásokat engedélyezzük, amelyről tudjuk, hogy használni akarjuk, és azt is, hogy milyen módon. A tűzfal és a belső gépek számára engedélyezett TCP/IP szolgáltatások mások lehetnek, de a belső gépeken is csak az okvetlenül szükséges dolgokat engedélyezzük. Kényelmi szolgáltatásokat semmiképpen. A belső gépek számára a tűzfalon át proxy szerverek segítségével biztosítunk kijutást. Belső hálónk forgalmát védjük az Internet felől jövő fenyegetésektől, legyen legalább egy szűrőnk.”*[12. p. 198.] Tehát a belső hálóról az Internet felé induló kéréseket is szűrjük meg. Átengedhetjük a *Web* (80, 8080, 443), az *ftp* (20,21), *mail* (*smtp, imap*, esetleg *pop3*), stb. munkát ténylegesen segítő portokat. Ha szükség van rá, akkor engedélyezzük az *ntp* (123/tcp/udp) protokollt, mely az idő beállításához használható. Ezt csak egy vagy két IP címről engedélyezzük, mert ez is támadási felület lehet, hiszen a naplófájlok emiatt össze-vissza írhatják az időpontokat. Az összes többi portot tiltsuk le a tűzfalon. (Az *irc* (6666,6667,6668), *napster*, és a különböző hálózati játékok portjait mindenképp tiltsuk le. Tiltsuk le továbbá a nem biztonságos szolgáltatások portjait, mint pl. a *telnet* (23).)

Azokat a tartalmakat, melyeket átengedünk, csak a tűzfalon engedjük ki, *proxy* (vagy *transzparens proxy*) segítségével. Enélkül az egész tűzfal semmit sem ér. A Web gyorsítására használjunk *Web-caching-proxy*-t (pl. *Squid*).

Mindenképp olvassuk el a *Firewall-Howto*-t.⁵⁹ Ajánlom továbbá a következő szakkönyveket: [16], [17], [18]. Két hasznos cikk a TCP működéséről és a hálózati biztonságról *Paul Russel*-től (az *ipchains* program írójától) [24], [25]. Elolvashatjuk ezenfelül a *Firewall Checklist*-et is.⁶⁰

⁵⁹ <http://metalab.unc.edu/pub/Linux/docs/HOWTO/Firewall-HOWTO>, A metalab magyar tükre: <ftp.fsn.hu/pub/docs/linux-howtos>

⁶⁰ <http://www.telstra.com.au/pub/docs/security/800-10/node69.html>

6. A Web-személyzet felépítése

A mai világban már annyira specializálódtak a feladatok, hogy egy ember nem képes átlátni egy ilyen komplex rendszert. Ezért van szükség a feladatok szétosztására.

Gyakorlatilag három rendszergazdára lenne szükség: Egy általános hardver/szoftver, egy adatbázis és egy Web-szerver rendszergazdára. (Sok esetben már a hardver és a szoftver gazdája is különválhat nagyobb mennyiségű gép esetén.)

A Web-személyzet ideális esetben a következő pozíciókból áll:

1. Rendszergazda: összeszereli és hálózatba köti a számítógépet, telepíti és karbantartja a szoftvereket, Az Ő feladata a biztonságos üzemeltetés és a biztonsági másolatok készítése is. Ő csak a Web-szerver és az adatbázis-szerver rendszergazdáknak ad belépési jogot a gépre. A „közönséges” felhasználóknak ideális esetben ezek a személyek adnak feladatuk szerinti hozzáférési jogot.
2. Adatbázis-rendszergazda: ő felügyeli az adatbázis szerver programot, ő hozza létre az adatbázis-felhasználókat (akik feltölthetik, módosíthatják az adatbázis tartalmát), ad nekik jogosultságokat az adatbázis elérését illetően. Hiba esetén a rendszergazdával együtt kell dolgoznia a helyreállítás érdekében.
3. Web-tervező: kialakítja a Web-hely arculatát, struktúráját, továbbá összefogja a fejlesztő-stáb munkáját. Ő a fő koordinátor, a többi taggal ő tartja a kapcsolatot.
4. Grafikus: a Web-tervező által kigondolt arculathoz grafikai terveket, munkákat és a konkrét fényképeket, képobjektumokat készíti el.
5. Web-programozó: az előbbi személyek által megtervezett dinamikus oldalakat programozza le valamilyen (pl. PHP3) script nyelven és azt a Web-tervező rendelkezésére bocsátja.
6. „Titkárnő”: ő viszi fel (gépeli be) a statikus szövegeket, melyeket a Web-tervező rendelkezésére bocsát. Felesleges egy képzett embert ilyesmire „kényszeríteni”. Továbbá ő lehet az, aki feltölti az adatbázist friss adatokkal, pl. termék és árlista. Neki, lehet, hogy egyáltalán nem kell felhasználói számlát létrehoznunk (Esetleg e-mail-t).
7. Webmester/webgazda (ált. a Web-szerver program rendszergazdája): elhelyezi és karbantartja a Web-lapokat, az ő feladata a website belső fájlstruktúrájának megtervezése, az e-mail-ek fogadása és megválaszolása (erre a témára vonatkozóan) esetleg továbbítása a megfelelő személyeknek.

Amint láthatjuk ezt nem sok kisvállalat engedheti meg magának. Ha már van egy rendszergazdánk (ekkor ő veszi át az összes rendszergazdai szerepkört), meg egy titkárnőnk, akkor a többit egy külsős cégre bízhatjuk. Ekkor meg kell bízunk a külsős cég alkalmazottaiban, mert azok a kész és a frissített anyagokat mindig fel kell hogy töltsék szerverünkre, tehát ide valamilyen hozzáférésük kell, hogy legyen. Ez persze nagymértékben csökkenti szerverünk védeltségét a külső behatolás ellen, hiszen nem

tudhatjuk, mennyire vigyáznak az ottani kollegák jelszavaikra. A másik lehetőség, hogy a friss anyagokat elküldik a rendszergazdának és az személyesen tartja karban a dolgokat.

A mi rendszergazdánk végzi a dolgát, és a titkárnőnk pedig frissítheti az adatbázist (pl. az árlistát) akár egy lekérdezős dinamikus Web-oldalon keresztül is.

7. Web-proxy fogalma és helye a hálózatban

A *Web-caching-proxy* egy olyan gyorsítótár-szerver, amely a Web-szerverekről lekért objektumokat tárolja. Ha egy új kérés érkezik, azt először a *proxy* kapja meg. Ha már megvan a tárolójában az adott objektum, és még érvényes annak a tartalma, akkor azt küldi el a kérőnek, és nem továbbítja a kérést a Web-szervernek. Főleg akkor hasznos, ha a belső hálózatunkat engedjük ki kliensként az Internetre. Ezt mindenképp egy jól beállított *proxy*-n keresztül érdemes megtenni, hogy a kisebb sávszélességű vonal terhelését csökkentsük. A Linux rendszereken a *squid* nevű GPL-es *Web-proxy* szoftver a legelterjedtebb.

A mi esetünkben a *Web-proxy* más jelentést kap. Ha nagyon nagy forgalmat bonyolít a szerverünk akkor a Web-szerver és az Internet kijárat közé beékelhetünk egy *Web-proxy*-t. Ez főleg a statikus tartalom esetében nagy terhet vesz le a szerverünk vállairól. Ekkor persze a külvilág a szervert nem láthatja, csak a *proxy*-t, vagyis *transzparens-proxy*-zást kell beállítanunk. A kliens azt hiszi a Web-szerverrel beszélget, közben csupán a *proxy*-val kommunikál.

Az Apache is rendelkezik egy beépített *proxy* modullal. Ez persze nem olyan hatékony, mint egy külön gép *squid*-et futtatva, de bizonyos esetekben és terhelési szinteken (talán) hasznos lehet.

8. Biztonsági alapok (hardver, szoftver)

A biztonság nagyon fontos, összetett és vitatható kérdés. Napjainkban egyre jobban előtérbe kerülnek a biztonsági problémák. A lényeg: nincs abszolút biztonság. Legfőbb ellenségünk maga a felhasználó (és a rendszergazda) lustasága. Ő az aki nem vigyáz kellően a jelszáva, ő az aki egy cetlire írja azt és ráragasztja a monitorjára, mert mindig elfelejti, stb. Egy hírhedt amerikai *cracker* elfogása után később elmesélte, hogy pofonegyszerű volt bejutnia a kormányzati rendszerekbe. Rendszergazdának adta ki magát és felhívott közalkalmazottakat, titkárnőket mindenféle ürüggyel, hogy azok adják ki jelszavaikat. Így nem is kellett szó szerint feltörnie a rendszereket, mert a kiskapu nyitva állt, onnan már szinte gyerekjáték volt a hiányos és átgondolatlanul megszervezett biztonsági kapukon átjutnia, hogy megszerezze a szükséges információkat.

8.1 Általános irányelvek

Az első pont tehát olyan rendszert tervezni, ahol a felhasználók a lehető legkevesebb kárt tehetik, vagyis a lehető legkevesebb, csakis az adott munkakörükhöz szükséges jogokkal szabad rendelkezniük.

A külső támadás lehetséges okai:

- A cégünk imázsának, jóhírének lerombolása
- Üzleti információk megszerzése, kémkedés a konkurenciának
- Unaloműző rosszindulatú károkozás
- Ugródeszka keresése más gépek feltöréséhez

A külső támadások fajtái:

- *Portscan*: az összes lehetséges portot végig próbálgatva ezzel méri fel a gépen futó szolgáltatásokat és, hogy azokat milyen programok nyújtják. Ezután el lehet dönteni, hogy a rendszer melyik része ellen kezdjenek támadást.
- *Sniffing*: Ha valakinek a belső (pl. *Ethernet*) hálózaton gépe van (vagy feltesz egy gépet) rendszergazdai jogosultságokkal (az egy-felhasználós rendszerek ilyenek), akkor ún. *promiscuous* módba kapcsolhatja a hálózati kártyáját. Ekkor minden csomagot, amely kikerül arra a fizikai hálózatra, a kártya elfogad magának is. Egy hallgatózó programmal felszerelve az illető megszerezhet bármilyen kódolatlan információt, amely „átfolyik” a kártyán.
- *DoS*: Itt a cél a szolgáltatás(ok) teljes lelassítása, megakasztása. Megfelelő (pl. TCP) kvótákkal szinte minden szolgáltatást meg lehet védeni ez ellen valamilyen mértékben.
- *DDoS (Distributed Denial of Service)*: Az előbbi támadásnak egy olyan fajtája, amikor sok „ártatlan” számítógépre valamely módon egy támadóprogramot telepítenek a tulajdonos engedélye nélkül, melyekről később egy időpontban indul „szétosztott” támadás a célgép ellen.
- *Spoofing*: DNS vagy IP cím átfedése. Pl. ha egy kliens lekérdez egy Web-oldalt, akkor egy ál név-szervertől nem a valódi Web-szerver címét kapja vissza, hanem előbb a károkozó gépét. Innen persze átirányíthat a kérés az eredeti szerverre, de közben akár hitelkártyaszámokat is lejegyezhet a *cracker* programja. Az IP átejtés esetében valaki úgy manipulálja a hálózatot, hogy a célgép IP címét ő veszi fel, és annak nevében kommunikál.

A belső támadások fajtái:

- *Buffer overflow*: egyes programok programozási hibáját kihasználva veremtúlsordulást okoznak, majd egy *rootshell*-t kérnek le az IP (*Instruction Pointer*) segítségével. Ekkor rendszergazdai jogosultságokat szerezhetnek, ha a megtámadott program (pl. *sendmail*) rendszergazdai jogkörökkel futott. Ez ellen

a programok biztonsági frissítésével lehet védekezni, és ha lehet alkalmazzunk „chroot jail”⁶¹-t és felhasználóként futtassuk a démont.

- *known temp file attack*: „E támadási módszer lényege: a behatoló megfigyeli, hogy a hibás program milyen néven hívja meg a .tmp file-jait, és abban a könyvtárban, ahol azokat eltárolja létrehoz egy ugyanolyan nevű szimbolikus linket, mint az egyik .tmp file. Ez arra a file-ra mutat, amit át akar írni.”[2]
- *Exploit-ok*: olyan előre gyártott programcskák, melyek minden különösebb szakértelem nélkül rendszergazdai jogköröket adhatnak az újdonsült *cracker*-nek. Ezek az Internetről le is tölthetők. <http://rootshell.org> Ilyen például a *rootkit* programcsomag, amely Linux alá is létezik. Ez trójai programokat tartalmaz, melyekkel kicserélve az eredetieket állandó kiskapu biztosítható a betörő számára.

Tehát a következő fontos elveket kell betartanunk:

- *Fizikai védelem*: a szervergép egy külön lévő elzárt, és jól zárható szobában üzemeljen, mely helyiségben (esetleg) légkondicionáló berendezés is van (~21 Celsius fok). A szobához csak az illetékeseknek legyen kulcsa. Takarító nem járhat a szobában, mert véletlenül kihúzhatja a vezetéket, stb. Továbbá fontos szempont a tűzvédelmi berendezés (pl. porral oltó), ráccsal védett ablakok, lehetőleg emeleti helyiség, stb. A mentés lemezeket egy másik helyiségben, esetleg épületben őrizni tűzkár miatt, jól elzárva, hogy illetéktelen személyek ne férjenek hozzá, stb. A helyiségben tilos a dohányzás, kávézás, stb.
- A gép BIOS-át úgy állítsuk be, hogy jelszóval legyen védve, csak a merevlemezről lehessen indítani, a felesleges hardvereket (floppy, soros, párhuzamos portok) tiltsuk le, ha nem használjuk őket.
- *Hivatalos biztonsági szabályzat* befoglalása a cég működési szabályzatába. A szabályok hatókörének és az egyes pozíciókon lévő alkalmazottak felelősségének megállapítása, stb.
- *Hálózati tervezés*: a hálózatot már a tervezéskor úgy kell kialakítani, hogy véletlenül se keveredjenek egy fizikai vagy logikai alhálózatra az egymással nem kapcsolatos szegmensek, kliensek.
- *Minimum elv*: csak azon programok / szolgáltatások fussanak és legyenek rajta a gépeken, amelyek ténylegesen és indokoltan használva lesznek.
- *Mindent tilos, kivéve, amit szabad elv*: a tűzfalon és más biztonsági szabályzó szűrőkön mindent letiltunk, és egyesével adjuk meg azokat a szabályokat, amelyek a „szabad” kategóriába tartoznak.
- *A jelszó rendelet*: minden jelszó legalább 8 karakteres, tartalmaz számokat és egyéb ASCII karaktereket és nem szótári szó. Nem tartalmaz semmi személyhez

⁶¹ chroot: Change Root, a gyökérkönyvtár átállítása futás előtt. Pl. a BIND/named démon képes arra, hogy felhasználóként fusson és becsukjuk egy alkönyvtárba. Ez általában az /etc/bind. Ha ide be is jut a betörő buffer overflow-val, a chroot börtönéből nehéz user-ként kijutni. Másik alternatíva a *BSD-s jail() függvény alkalmazása.

köthető, pl. valaki születésnapját, stb. A jelszavak 30 naponként változnak meg, stb. (Használjunk MD5 kódolású jelszavakat, lásd később). Használjunk *shadow password* funkciót.

- *Email titkosítás*: amikor csak lehet, a szerveren lévő felhasználók kódolt formában levelezzenek és a rendszer is legyen képes kódolt levéltovábbításra. Ehhez már megvannak a szükséges programok: felhasználói oldalon a `gpg`⁶², vagy a `pgp`⁶³, szerver oldalon a TLS-képes⁶⁴ levéltovábbító, mint pl. a `postfix-tls`. Továbbá a rendszer eseménynaplóját is titkosított formában ajánlott elküldeni a rendszergazdának, hogy ez se legyen lehallgatható.
- *Őrizetlen terminál*: A felhasználók ne hagyják őrizetlenül a számítógépet bekapcsolt és bejelentkezett állapotban, mert bárki leülhet mellé és kárt tehet. Használjuk a `vlock` vagy `xlock` programokat a terminál lezárására.
- *Csak az kapjon shell-t akinek arra tényleg szüksége van*. Akinek a munkájához nem szükséges a UN*X *shell* használata, pl. csak a böngészőjén keresztül módosítja az adatokat, annak ne legyen *shell*-je.
- Nem biztonságos szolgáltatások (pl. `telnet`, `ftp`) tiltása és helyettük kódolt változataik beállítása (`ssh`, `scp`)
- Lehetőséget adni a Web-szerver látogatójának a titkosított adatcserére (SSL) Érzékeny fájlokat / könyvtárakat azonosításhoz kötni.
- *Folyamatos mentés*: adott időközönként teljes vagy részleges mentést kell végrehajtani a rendszerről, hogy hiba esetén gyorsan visszaállítható legyen az utolsó működő állapot. Egy mentési szabályzatot is ki kell dolgozni: milyen időközönként milyen mértékű mentést kell végezni, visszamenőleg hány állapotot kell megtartani, stb.
- *Rendszernaplók nyomon-követése*: a rendszergazdának figyelnie kell a rendszer eseményeit, és ha valami kompromittálásra utaló jelet találna, akkor azonnal cselekednie kell.
- *Alapos ok nélkül semmilyen programot újabb verziójúra cserélni nem szabad. Egy jól működő rendszerhez hozzányúlani nem szabad*. Csakis a biztonsági problémák miatt szabad új verziókat feltenni. Ha mégis lecserélni szándékoznánk valamit, csak már egy előre letesztelt, kipróbált változatot tegyünk fel. Tudniillik „a pokolba vezető út is jószándékkal van kikövezve”. Mi jószándékkal feltelepítjük a legfrissebb verziót, amely lehet, hogy teljesen másképp fog viselkedni, és akkor állhatunk neki tanulmányozni a hiba okát. Mindazonáltal, nem szabadna ~2 évnél idősebb szoftvert használni. Van néhány olyan hiba, amit a fejlesztők kijavítanak, de nem publikálnak. Ésszel kövessük mindig a legfrissebb stabil verziókat.

⁶² GNU Privacy Guard

⁶³ Pretty Good Privacy, ez jobbára kereskedelmi program.

⁶⁴ Részletesen a IV. Megvalósítás / 2. Finomítás / 2.3 A levelező démon beállítása / A titkosítás beállítása fejezetben.

- A „nagy” levelező szerveren sose legyen Web-szerver és *vica versa*. Általában a `sendmail` program a legérzékenyebb a betörésekre és rajta keresztül elérhető lehet a Web-szerver. Miután az OpenBSD csapat elkezdte auditálni a kódját, sokat javult a biztonsága, de azért Én az életemet nem bízom rá. A levelezőszerveren használunk egy sokkal biztonságosabb alternatívát, mint amilyen például a `postfix` vagy a `qmail`.
- Ne futtassunk FTP szerveret a Web-szerveren. Szintén sérülékeny pont a *buffer overflow* szempontjából. Lehetőleg a legfrissebb, legstabilabb szerveret használjuk. Ha lehet, ne legyen *Anonymous* (névtelen) *ftp*, de ha mégis, akkor vigyázzunk a jelszófájltra. Ne engedjünk meg olyan könyvtárat, amelyben *Anonymous* olvashat és írhat is.
- A Web-szerver démon sose fusson *root* jogokkal, kapcsoljuk ki a könyvtárlistázást, és ne kövesse a szimbolikus linkeket. (Ezekről részletesen később)

Ha a rengeteg jelszót nem tudjuk megjegyezni, ne használjunk ugyanolyan jelszókat több helyen, ne írjuk le őket papírra, se kódolatlan fájlba. Használjuk pl. a `gpasman` programot. (<http://gpasman.nl.linux.org>) Ez a program tipikusan a sok jelszó menedzselését segíti. A jelszavakat kódolt formában tárolja. A Woody-ban már benne lesz. Töltsük le a rendszergazdai gépre (vagy fordítsuk le forrásból). Többben mindenféle kézigépekbe írják a jelszavaikat. Fontos, hogy ez esetben egyrészt kódoljuk az eszközben az információt lopás esetére, másrészt pedig tartsunk róla biztonsági másolatot, ha a készülék elromolna (ellopják, elvész), ne kelljen mindent előről kezdeni.

Sokan viszont az javasolják, hogy semmiféle jelszót ne tároljunk hálózatra csatlakoztatott gépen, hiszen valamiképpen ekkor úgyis hozzá lehet jutni. Persze ekkor meg kellene jegyezni az összes jelszót, amire kevés ember képes.

Fontos feladat a rendszergazdának a rendszer folyamatos frissítése a biztonsági javításokkal, a biztonsággal foglalkozó oldalak látogatása. Pl. <http://www.linuxsecurity.com>, <http://www.debian.org/security>, <http://www.cert.org>. Olvassuk el a *Compromise FAQ*-t (<http://www.iss.net>), a *Linux Security- HOWTO*-t <http://metalab.unc.edu/pub/Linux/docs/HOWTO/Security-HOWTO> A tűzfal és a kernel tűzfalfunkcióját szabályzó `ipchains` program HOGYAN-ját ugyanitt találjuk [Firewall-HOWTO](#) és [Ipchains-HOWTO](#) néven. Az utóbbi magyar fordítása: <http://www.rkcs.hu/linux/index2.html> A <http://www.faqs.org/rfcs/rfc2196.html> címen egy biztonságpolitikai szabályzatot találunk RFC-be foglalva. Továbbá érdemes áttekinteni az 1244 és 1281-es számú RFC-eket is, melyek szintén ezzel a témával foglalkoznak. Végül egy valós életbeli példa található az <ftp://coast.cs.purdue.edu/pub/doc/policy> címen.

Ajánlom továbbá az olvasó figyelmébe a következő könyvet: [12]

8.2 A Linux kernel biztonságát növelő projektek

A Linux-hoz létezik több biztonsági „folt” is. Az egyik ilyen érdekes és hasznos kód a <http://www.openwall.com/linux> könyvtárban található. Jelenleg csak a 2.0 és 2.2-es sorozatú kerneleket támogatják. Sok biztonsági javítás kerül később innen a hivatalos kernelforrásba. A következők ellen nyújt bizonyos mértékű (nem 100%-os) védelmet:

- Nem futtatható felhasználói veremterület védelme a puffer túlcsordulásoktól.
- Szimbolikus link és FIFO korlátozás a `/tmp` könyvtárban
- `/proc` könyvtár információinak védelme a felhasználói kutakodástól és információgyűjtésről (csak az adott csoportba tartozó emberek tekinthetik meg a fájlok tartalmát)
- A fájl leíróablák (*File descriptors*: 0,1,2) speciális kezelése
- A felhasználók által maximálisan futtatható folyamatok számának hatékonyabb korlátozása
- Használaton kívüli megosztott memória szegmensek megsemmisítése (kinullázása)

Hasznos megfontolni ennek a foltnak a használatát, hiszen növelheti a rendszerünk biztonságát. Hátránya viszont az, hogy megfelelő tesztelés kell, hogy megelőzze a használatát, mert egyes „veszélyes” módon viselkedő programok esetleg nem fognak rendesen futni. (Tapasztalatom szerint minden jól működött.) Ha használni szeretnénk, mindenképp olvassuk el a dokumentációját, hogy megértsük, miről is van itt szó és milyen szintű biztonsági erősítést kapunk ezáltal. Ezt a foltot a kezdőknek is ajánlom, mivel nem igényel semmilyen különös karbantartást.

Meg kell említenem a **LIDS** (*Linux Intrusion Detection System Patch*, vagyis Linux Betörés Detektáló Rendszer) foltot is. (<http://www.lids.org>) A LIDS képes együttműködni az OpenWall folttal és erős biztonsági rendszert épít a Linux-ba. Lényegében ez abban a fázisban fontos, amikor már valaki behatolt a rendszerbe és *root* jogokat szerzett. Ez a program lekorlátozza a *root* jogait. Amikor nekünk kell adminisztrálni a gépet, egy jelszó megadásával kikapcsolhatjuk a védelmet, hogy tudjunk dolgozni.

Korlátozások:

- Megtiltja a modulok betöltését és eltávolítását.
- Megtiltja a közvetlen memória-kezelést
- Megtiltja a közvetlen lemez-kezelést
- Megtiltja a közvetlen I/O port-kezelést
- Védi az indulási folyamatban résztvevő fájlokat (kernel, *lilo*, démonok, modulok, init script-ek)

Behatolás-figyelés:

- Naplózza a tiltott dolgokhoz való hozzáférési próbálkozásokat
- Csak olvashatóvá ill. csak hozzáfűzhetővé teszi a naplófájlokat, hogy a behatoló ne tudja eltüntetni nyomait
- Elrejtí a behatolás-figyelő program részeit

Rendszer-védelem:

- Az útválasztó-táblák és a tűzfal-szabályok védelme
- A felfűzések (*mounts*) befagyasztása
- A démonok védelme a szignáloktól (pl. `kill`)
- Kernel jogosultságok – a *root user* szintre butítható, stb. (bővebben a dokumentációban)

A rendszert egy `lidsadm` nevű démon kezeli, melyhez egy *RipeMD-160* kódolású jelszóval lehet csak hozzáférni. Ez a program monitorozza az eseményeket. A védelmet rajta keresztül lehet ki és bekapcsolni. Külön védelem állítható be szinte mindenhez a rendszeren.

Ha érdekel minket a dolog, olvassuk el a dokumentációt. A lids@egroups.com címen érhető el a rendszer levelező listája. A függelékben egy részletesebb útmutatóban tárgyalom a beállítását. Ezt a programot a haladóknak ajánlom.

A következő fontos és hasznos törekvés a **Medusa**, melyet Szlovákiában fejlesztenek - többek között szlovákiai magyarok is. Ez a programcsomag kernel foltból és egy démonból áll. A cél kernel szintű felhasználó azonosítás. Ez az „azonosító-szerver” átlátszóan működik a kernel és a felhasználói programok között. Bizonyos műveletek indítása előtt a kernel jóváhagyást kér a szervertől. Ezzel a módszerrel szinte bármilyen biztonsági modell megvalósítható. A konfigurációs fájlok megfelelő beállításával nagyon magas szintű biztonságot érhetünk el. A kernel és a démon egy speciális `/dev/medusa` eszközön keresztül kommunikál.

A jelenlegi implementáció a következőkre képes:

- Teljes hozzáférés szabályozás (*Access control*) a fájlrendszeren
- Hozzáférés átirányítása egyik fájlról a másikra
- A jel (*signal*) küldés / fogadás teljes szabályozása
- Fontos folyamat-műveletek direkt irányítása
- Bármely rendszerhívás indításának szabályozása egy adott folyamaton belül
- Egyes fájlok és/vagy folyamatok teljes elrejtése más folyamatok elől
- Minden folyamat saját bejelentkezési azonosítót kap
- Adott kód végrehajtásának kikényszerítése
- Bármely rendszerhívás alacsonyszintű szabályozása

Hátránya, hogy egyelőre csak Intel architektúrán működik, de már folyamatban van a kód portolása más rendszerekre is. A tesztek szerint jól működik többprocesszoros rendszereken is. A másik nehézség a rendszer beállítása, egy kis C nyelvi programozói múlt nem árt hozzá. A forráskód letölthető a <http://medusa.fornax.sk> címről. Amennyiben érdekel bennünket a dolog, olvassuk el az egész dokumentációt és kövessük az ott leírtakat. Segítséget kérhetünk a csapat levelezőlistáján: medusa@medusa.fornax.sk

Azt, hogy a Medusa együttműködik-e az előzőekkel, sajnos nem tudom és nem is garantálhatom. (Van egy-két ember a levelezőlistán, aki már készített vegyes foltokat, melyek több biztonsági foltot együtt tartalmaznak.) Végül is, egy jól felépített / beállított *Medusa* nagyrészt feleslegessé teheti a másik két kódot.

Én az OpenWall-féle kódot alkalmazom a mintapéldámban. A függelékben röviden bemutatom a LIDS féle rendszert is.

8.3 A Web-alkalmazások biztonsága

Nem egy Web-helyet a hibásan elkészített Web-alkalmazásokon keresztül törnek fel, vagy férnek hozzá egyes felhasználók személyes adataihoz. Ezeket a hibákat a Web-programozónak kell kiküszöbölnie a kódok rendszeres ellenőrzésével. Röviden bemutatom a betörési technikákat:

- „Süti mérgezés”: A felhasználó gépére a Web-szerverről kisebb, a későbbi azonosításhoz szükséges információkat tartalmazó fájlok kerülhetnek. Ezeket *cookie*-knak, magyarul sütiknek nevezzük. Ezeknek két fajtája van: egy állandó, azaz a lemezen lévő, és egy nem-állandó, vagyis a memóriában lévő süti. Természetesen a legtöbb kliensgépen a kódolatlan szöveges állományokként helyet foglaló sütik könnyen elolvashatók mások számára. Ezek használatát kerüljük. Továbbá a kódolatlan sütik hálózati szaglászással is elfoghatóak. Ezért javasolt az SSL csatorna használata kényes információkat tartalmazó sütik esetén. Sok szakember teljesen ellenzi a sütik használatát azok megbízhatatlansága miatt. Az elkapott sütik segítségével jelszavak és hitelkártyaszámok is megszerezhetők.
- Űrlap manipulációk: A károkozó letöltve egy űrlapot végignézheti annak HTML kódját és azt módosítva küldheti vissza. Általában több elemet tartalmaz, mint amit az értelmező vár, ezzel pl. *buffer overflow* támadást indíthat a rendszer ellen. Más esetekben parancsok elindítását kezdeményezhetik a szerveren. Ha az értelmező script *root*-ként futott, máris kész a bejárat. Védekezés: ellenőrizni kell az űrlap integritását értelmezés előtt, továbbá a kapott mezők értékeit nagymértékben szűrni és ellenőrizni kell a szerveren végrehajtás előtt.
- Több űrlap esetén egyes űrlapok kihagyása: Ha több űrlapot kell kitölteni egymás után, a károkozó személy direkt URL begépelésével kihagyhat néhány lapot, ezzel érvénytelen adatbázis rekordokat generálhat. Biztosítani kell, hogy csak akkor dolgozzon fel az értelmező egy adatsort, ha minden űrlap ki lett töltve.

- Direkt adatbázis lekérdezések: Sokszor az űrlap mezőiből direkt lekérdezések generálódnak az adatbázis felé, melyre a választ a következő oldal hozza. Ha a károkozó ismeri a programnyelvet (pl. SQL), akkor más felhasználókra vonatkozó információkat is lekérdezhet. Megoldás lehet itt is a mezők szűrése és az adatbázis hozzáférési jogosultságainak helyes beállítása.
- Könyvtárlistázás: Ha olyan könyvtárak is listázhatóak, melyek a Web-alkalmazás kódját tartalmazzák, akkor a kód könnyen megszerezhető és abban biztonsági hiba kereshető ki. A CGI-ket, PHP, stb. kódokat tartalmazó könyvtárak semmiképp se legyenek listázhatóak. (És limitáljuk a fel/letöltésüket.)

Bővebb információkért nézzük át a szakirodalmat: [26], [27], [28].

9. SSH - Távoli menedzsment

Amint a rendszer változtatást, karbantartást igényel, a rendszergazda nem szaladgálhat be állandóan a lezárt szerver-helyiségbe, pláne, ha otthon ül, vagy épp úton van több száz kilométerre a szervertől. De ekkor is ki kell javítani az esetleges hibákat, ellenőrizni kell a rendszert. Ezért be kell jelentkeznie egy távoli gépről a szerverre, hogy elvégezze a karbantartást. Ha szerver nem állt le teljesen, van áram és a bejelentkezéshez szükséges minden hálózati kapcsolat él, akkor semmi akadálya, hogy a rendszergazda bejelentkezzen. Persze ezt a hagyományos `telnet`⁶⁵ programmal is megtehetné, de hát bolond lenne, hiszen valahol talán egy lehallgató program pont erre vár, hogy a beírt rendszergazdai jelszavát elmentse és elküldje valakinek. Ezért a rendszergazda az `ssh` (Secure shell – biztonságos héj) program segítségével lép be rendszerébe. A szerveren futnia kell egy ún. `sshd` démonnak (szolgáltatás, kiszolgáló) és a kliens gépen kell lennie egy `ssh` kliensnek. Ez szinte minden platform alá létezik. BSD licenz alatt megjelent egy szabad forráskódú implementáció (<http://www.openssh.com>)⁶⁶. Egyébként a sokkal szigorúbb licenszel rendelkező SSHv1 és SSHv2 szerver/kliens csomagokat is választhatjuk.⁶⁷

„A hálózatba kötött gépek távoli kezelése egyszerűen megoldható. Akár telnetes kapcsolaton keresztül (nem biztonságos), akár ssh vagy ssl segítségével (ezek már biztonságosak). Ezek karakteres felületeket biztosítanak, így egy lassú kapcsolaton (modem) keresztül is alkalmazhatóak. És hozzá kell tenni azt is, hogy egy profi rendszeradminisztrátor sokkal gyorsabban végzi a munkáját parancssorból, mint grafikus felületen. Ezenkívül léteznek karakteres terminálon is futtatható programok, amelyek menükön keresztül kommunikálnak a felhasználóval.

⁶⁵ Eredetileg a nagy Mainframe gépekre jelentkeztek be terminálokról. A telnet egyfajta terminál-emulátor.

⁶⁶ Ezt az OpenBSD operációs rendszer keretében fejlesztik, nemsokára belekerül a ssh v1.5 és v2 protokollok implementációja is. Azért .com a web-hely címe, mert valaki már bejegyezte előttük a <http://www.openssh.org> címet és ott más anyagokat helyezett el.

⁶⁷ A választható csomagok táblázatát az 4. Alternatívák a távoli bejelentkezésre c. fejezetben találhatja az olvasó.

*Megfelelő beállítások esetén a rendszergazda közvetlenül is bejelentkezhet az adminisztrálni kívánt gépre, akár grafikus felületen keresztül is, anélkül, hogy zavarná a gépen dolgozó egyéb felhasználókat. Szinte minden változtatást végre lehet hajtani a gépen, az operációs rendszer újraindítása nélkül is.*⁶⁸

*„A távoli felügyelet lehetőségeinek köszönhetően a rendszergazdának el sem kell mozdulnia a számítógépe mellől ahhoz, hogy karbantartsa a gondjaira bízott gépeket. Az átlagos felhasználó teljesen a saját képére formálhatja az általa használt rendszert, és ehhez az operációs rendszeren nem kell változtatni, nem kell elállítani semmit sem. Ily módon a működőképesség fenntartása nem kerül erőfeszítésbe, mert nem szükséges változtatni a jól beállított rendszeren. A biztonsági rendszer miatt pedig a felhasználó nem tud elállítani semmit a gépen, amihez nincsen joga.*⁶⁹

A tapasztaltabb rendszergazdák régóta tudják, hogy parancssoros üzemmódban sokkal hatékonyabban lehet dolgozni és karbantartani, mint mindenféle grafikus felületű ikonok és menük rengetegében.

Azoknak, akiknek mégis fontos a grafikus felület segítsége, ott a `linuxconf` programcsomag. Ezzel szöveges módban egy menüs programmal szerkesztgethetjük a legfontosabb konfigurációs állományokat. Létezik hozzá egy X11 grafikus felületű (`linuxconf-x11`) kezelő is, ha kell. Ezt a csomagot telepítve a szerveren, a mi távvezérlő gépünk X szerverén meg fog jelenni a program, (javasolt egy `ssh socket`-be becsomagolni!) és máris állíthatunk kedvünkre.

Összegezve, ma már a karbantartás elképzelhetetlen távoli bejelentkezés nélkül. További információkat a IV. *Megvalósítás / 2. Finomítás / 2.4 Az SSH konfigurációjának finomhangolása* és az VII. *Alternatívát nyújtó programok a Debian-ban / 4. Alternatívák a távoli bejelentkezésre* c. fejezetekben találhat az olvasó.

10. PHP3 alapok (dinamikus Weblap-készítés)

A PHP egy szerveroldali értelmező script nyelv. A PHP nyelvet Rasmus Lendorff készítette először. Később rengeteg programozó beszállt a fejlesztésbe, ahogy a nyelv egyre népszerűsödött. Miután a PHP alapjait teljesen újraírták, megszületett a PHPv3⁷⁰.

Tulajdonságai:⁷¹

- Nyitott forráskód, GPL licenz
- Szerveroldali, nem kíván speciális böngészőt

⁶⁸ <http://www.vmg.sulinet.hu/vmg/home/szamtech/linux/node22.htm>

⁶⁹ <http://www.vmg.sulinet.hu/vmg/home/szamtech/linux/node23.htm>

⁷⁰ Azóta már megjelent a PHPv4 is, melyben a PHP egy részét megint újraírták. Részletesen később.

⁷¹ [3] p. 2-3 alapján

- Többplatformos
- Több mint 600 ezer Web-helyen használják⁷²
- HTML-be ágyazott
- Egyszerű szintaktika
- Kis erőforrásigény – az Apache moduljaként futhat
- XML kezelése
- Adatbázis kapcsolat mind szabad, mind kereskedelmi adatbázis-szerverekhez
- Fájlkezelő rutinok
- Szövegkezelő rutinok
- Sokféle változó, komplex adatszerkezetek lehetősége
- Képfeldolgozó rutinok – dinamikus képek előállítása
- és még sok más

Ez egy HTML-be beépülő nyelv, mely a szintaxisának egy részét a C, Java és Perl nyelvekből vette át. A PHP magas szinten integrálva van az Apache Web-szerver programba. Emiatt sokkal gyorsabb az Apache-PHP páros, mint az Apache-PerlCGI, mivel nem kell külső értelmezőt indítani.⁷³ A másik fontos tényező az, hogy mivel szerveroldali a kód értelmezése, ezért a végfelhasználó a böngészőjében már csak HTML kódot kap, minden PHP kód HTML-lé lesz fordítva. Így egyrészt nem kell a böngészőnek az értelmezéssel törődni (mint pl. Java), másrészt az értékes munka, a Web-programozói kód sem kerül ki a szerverről és azt más így nem használhatja fel.

A PHP3 hátrányai: „

- *Az Apache (tehát egyúttal a modulok, így a PHP3 is) mindvégig ugyanazon felhasználó jogaival fut [...] CGI esetén suEXEC vagy setuid bit segítségével el tudjuk érni, hogy a script biztonságosan a mi jogainkkal fusson. [...] Ezt azonban az Apache-on belül (a Unix biztonsági rendszerének felépítéséből következően) nem tehetjük meg – így vagy osztoznunk kell, vagy új process indítására kényszerülünk (lásd php3-cgi), de ezzel elvesztjük a PHP3 legnagyobb előnyét, a kezdési időt. A PHP3 tehát akkor ideális, ha az adott szerveren csak egyvalaki (például a webmaster) vagy egymással teljes mértékben együttműködők dolgoznak.*
- *A hosszabb, számításigényes feladatok lassan futnak, mivel a PHP3 utasításértelmezője lassú⁷⁴ [...] Bonyolultabb feladatoknál érdemes áttérni Perl-re, vagy C-re.*
- *[...] az Apache több példányban fut egyszerre, és ez megnehezíti a letöltések közti adatmegtartást. Bár ez a probléma kis kényelmetlenség árán megoldható, de akinek*

⁷² [3] p. 5

⁷³ Azóta már létezik Apache-ba integrált Perl értelmező modul is, így szorosabbá vált a verseny.

⁷⁴ Ezt a PHP4-ben nagyrészt kiküszöbölték. Ha ilyen feladattal találkozunk, váltsunk inkább PHP4-re.

feltétlenül megmaradó adatokra van szüksége, annak a Roxen Challenger Web-szerveret ajánljuk.⁷⁵ [13]

Ugyanúgy mint az Apache, a PHP is szét van darabolva különböző csomagokra annak érdekében, hogy csak azt kelljen felrakni, amire igazán szükség van. Ezáltal kevesebb erőforrást foglalunk le.

php3	Az alapcsomag. Ez tartalmazza a betölthető modulokat az Apache szerverhez, néhány extra funkciót nyújtó modult és a php2-php3 konvertert.
php3-doc	Az Online dokumentációkat tartalmazza.
php3-dev	A fejléc fájlokat tartalmazza (új modulok fordításához.)
php3-gd	E modulnak a segítségével dinamikus grafikákat készíthetünk (a libgd grafikus könyvtárát használva).
php3-imap	IMAP ⁷⁶ függvények hívása közvetlenül PHP script-ből.
php3-ldap	LDAP ⁷⁷ funkciók meghívása közvetlenül PHP script-ből.
php3-magick	ImageMagick ⁷⁸ funkciók meghívása közvetlenül PHP script-ből.
php3-mhash	MHASH ⁷⁹ funkciók meghívása közvetlenül PHP script-ből.
php3-mysql	MySQL kapcsolat létrehozása közvetlenül PHP script-ből.
php3-pgsql	PostgreSQL kapcsolat létrehozása közvetlenül PHP script-ből.
php3-snmp	SNMP ⁸⁰ funkciók meghívása közvetlenül PHP script-ből.
php3-xml	XML ⁸¹ kezelő funkciók meghívása közvetlenül PHP script-ből.
php3-cgi	Egyedülálló (Apache nélküli) értelmező. Ekkor más Web-szervereket is használhatunk ⁸² . Minden fenti modul megtalálható CGI-s változatban is. Ezek az Apache-al is használhatóak, de akkor a sebesség kisebb lesz, viszont a futtató felhasználói jogkör megváltozhat.

3. táblázat - PHP3 csomagok a Debian-ban

Ha az Apache fut és php3 modul be van töltve, akkor egy egyszerű kis programcskával letesztelhetjük. Írjuk be ezt a *shell prompt*-ba:

```
echo "<?php phpinfo() ?>" > /var/www/phptest.php3.
```

Ezután ízlés szerint kedvenc böngészőnkkel tekintsük meg az oldalt, pl.:

```
lynx http://localhost/phptest.php3
```

⁷⁵ Lásd: VII. Alternatívát nyújtó programok a Debian-ban / 1.1 Roxen

⁷⁶ Internet Mail Access Protocol: levelezéskor használható levélküldő és fogadó protokoll.

⁷⁷ Lightweight Directory Access Protocol: ez egy címtár-szolgáltatást nyújtó protokoll.

⁷⁸ Ez egy raszterkép-manipuláló programcsomag és függvénykönyvtár.

⁷⁹ Ez a titkosítást és az autentikációt segíti. „Hash” képezhető, pl. MD5 vagy SHA1 eljárással. (Nem fordítják magyarra.)

⁸⁰ Simple Network Management Protocol: Hálózati eszközök felügyeletét végzi. Segítségével intelligens hálózati megfigyelő rendszer létesíthető.

⁸¹ eXtensible Markup Language: a HTML-t felváltó, újgenerációs leíró nyelv.

⁸² A PHP4-et már integrálták a Roxen-be is. Lásd később.

Ha minden jól be van állítva, akkor itt egy hosszú információs oldal keletkezik, amely a szervergép és a rajta futó *Web* és *PHP* programok adatait listázza ki.

A továbbiakban bemutatok egy egyszerű programrészletet ízelítőnek. A programozás részletes bemutatása meghaladja e munka kereteit. Javaslom az Online dokumentáció és a szakirodalom (pl. [3]) tanulmányozását a Web-programozásban érdekelteknek. Információk a hivatalos Web-oldalon bőven fellelhetők: <http://php.net>. Néhány hasznos tipp és trükk: http://phpclub.unet.ru/index_e.php3

A következő egyszerű programocska a „Hello világ!” PHP-s megvalósítása.⁸³ Természetesen ez korántsem mutatja meg a PHP erősségeit. Három fájlt készítünk. Az első egy függvény (*include*) fájl. Innen rutinokat hívhatunk meg – nem kell megírni minden egyes .php⁸⁴ fájlba egy adott függvényt.

Az első fájl két függvényt tartalmaz. Az első az oldal címét változtatja meg, a második pedig számokat ír ki ciklus segítségével.

/var/www/hello.inc:

```
<?php
function printtitle()
{
    print "<title>Helló a hello.inc fájlból.</title>\n";
}

function printnumbers($start)
{
    print "<H2>";
    for($temp=0; $temp < 5; $temp++)
    {
        print $start++ . "<br>\n";
    }
    print "</H2>";
}
?>
```

Ez a fájl egy HTML fájl, mely PHP kódot tartalmaz. Ebből hívjuk meg az előző hello.inc fájlt, hogy kiírjuk a lap címét.

/var/www/hello.php3:

```
<HTML>
<?php include("./hello.inc") ?>
<HEAD>
<?php printtitle() ?>
<META HTTP-EQUIV="pragma" CONTENT="nocache">
</HEAD>
<BODY>
```

⁸³ http://www.troubleshooters.com/tpromag/200004/200004.htm-hesh_part2#_phpprogramming alapján

⁸⁴ Nem kötelező a .php3 kiterjesztés használata. Bármilyen tetszőleges nevet használhatunk.


```
A szövegtest kezdete<p>
<?php
printnumbers(7);
?>
<p> A szövegtest vége<p>
</BODY>
</HTML>
```

Ha futtatjuk az oldalt egy böngészőben (jelen esetben a lynx-ben) akkor a következő képet kapjuk vissza:

```
Helló a hello.inc fájlból.

A szövegtest kezdete

7
8
9
10
11

A szövegtest vége
```

A Potato-ban a PHP csomagok karbantartója *Madarász Gergely*. A PHP dokumentációjának magyar fordítása letölthető innen: <http://weblabor.hu/php>
A Debian Potato-ban 33 csomag foglalkozik a PHP3-al, 14 pedig a PHP4-el (Lásd később).

11. MySQL alapok (adatbázis szerver)

A MySQL egy igazi többfelhasználós, többszálúsított SQL adatbázis szerver. Jelenleg az SQL (*Structured Query Language*, Strukturált Lekérdező Nyelv) a legelterjedtebb és szabványos adatbázis nyelv világszerte. A rendszer kliens/szerver felépítésű.

A MySQL legfőbb erényei a gyorsaság, robusztusság és a (viszonylag) könnyű használat. 1996-ban kezdték fejleszteni a T.c.X. nevű cégnél., ahol azóta több mint 40 adatbázisban tárolnak 10000 táblát, melyből csak 500-ban 7 millió sor van. Ez kb. 100GB adat.

A MySQL-t a <http://web.mysql.com> hálószenen érhetjük el. Itt található Online dokumentáció, melynek nagy része természetesen benne van a Debian-ban is.

A MySQL sajnos nem teljesen szabad szoftver⁸⁵. Saját licenszpolitikájuk viszont megengedi ingyenes használatát sok platformon és szituációban. A mi esetünk: „ 3.5.4 Running a web server using MySQL: If you use MySQL in conjunction with a web server

⁸⁵ A legfrissebb változata már teljesen GPL-es, de ez nem fog belekerülni a Potato-ba. A Woody változat tartalmazni fogja.

*on Unix, you don't have to pay for a license. This is true even if you run a commercial web server that uses MySQL, because you are not selling MySQL itself.”*⁸⁶

Vagyis, ha egy Linux-os Web-szerveren futtatjuk, legyen az akár üzleti célú is, számunkra ingyenes a használata.

Licenszet akkor kell vásárolni, ha:

- Eladjuk a MySQL szervert egy másik termék vagy szolgáltatás részeként.
- Pénzt kérünk a MySQL telepítéséért és üzemeltetéséért valakitől.
- Beletesszük egy olyan terjesztésbe, amiért pénzt kérünk és az nem terjeszthető ingyenesen tovább.
- Nem UN*X platformon futtatjuk / használjuk.

Ekkor licenszet kell venni minden olyan gépre, amin a szerver fut. Az egyik kliens kódja GPL alatt van, ezért arra ezek nem vonatkoznak.

Itt [8] egy hasznos olvasmány a papíralapú dokumentációt kedvelőknek. Ezek [9],[10],[11] pedig az SQL nyelvet taglalják.

A következő címen MySQL + PHP mintapéldákat találhatunk:

<http://www.wernhart.priv.at/php>

libmysqlclient6	3.22.30	a kliens oldal függvénykönyvtára
libmysqlclient6-dev	3.22.30	fejléc fájlok fejlesztőknek
mysql-gpl-doc	3.22.30	Online dokumentáció GPL licenz alatt info, HTML, és text formátumban
mysql-gpl-client	3.22.30	GPL-es kliens binárisok
mysql-manual	0.95	Mike Miller nemhivatalos kézikönyve. Ez a non-free szekcióban található. Már idejétmúlt, de hasznos lehet
mysql-doc	3.22.32	a non-free Online dokumentáció
mysql-client	3.22.32	a non-free kliens binárisok
mysql-server	3.22.32	az adatbázis-szerver motorja
www-mysql	0.5.7	Web-interfész – segítségével SQL parancsok építhetők be a Web-oldalakba. A parancsok a szerveren hajtódnak végre és az eredményt HTML-ben küldi el a felhasználónak ⁸⁷
xmysqladmin	1.0	egy frontend (kliens) az adatbázis motorhoz. X11 grafikus rendszerekben használható, funkciói: a szerver újraindítása, státusz ellenőrzés, folyamat ellenőrzés, jogosultságok kezelése, adatbázisok / táblák kezelése

⁸⁶ http://web.mysql.com/manual/Licensing_and_Support.html

⁸⁷ Ez egy alternatív út a php3-mysql helyett. Én az egységes programozás miatt a php-s megoldást javaslom.

4. táblázat - MySQL csomagok a Debian-ban

A szerverre nem elég feltenni a `mysql-server` csomagot. Ha ott akarjuk kezelni az adatbázisokat / táblákat is `ssh` segítségével, akkor valamelyik klienst is fel kell tenni. Érdekes lehet egy távoli gépről karbantartani az `xmysqladmin` segítségével is, mely könnyen kezelhető grafikus megoldást kínál. Ez esetben a programot telepítsük inkább a távoli gépre. Ha nem a szerveren végezzük a feltöltést, akkor a `mysql` portját is engedélyeznünk kell a megfelelő hálózatok felé. Ez azonban elég veszélyes is lehet. Javaslatom az, hogy egy jól megírt PHP-s programmal tartsuk karban az adatbázist a Web-szerveren keresztül. (SSL és felhasználó-azonosítás segítségével) Ekkor a `mysql` csak a szerveren belül lesz (legyen) elérhető.

A MySQL hibaüzeneteinek egy része már le van fordítva magyar nyelvre is.

III. Tervezés

A cégnél tehát összeül a döntéshozó és a szakember, hogy megbeszéljék a rendszer tervét. A vezetőség kifejti elképzeléseit, igényeit a rendszerrel szemben felhasználói szemszögből. A rendszergazda ennek alapján összeállítja a hardver és az Internet / hálózat tervét, majd költségbecslést készít. A cég megjelöli, hogy milyen domén-neveket kíván bejegyezni. A rendszergazda, vagy a leendő Internet szolgáltató bejegyezteti a domén-neveket. (Esetünkben a *boresszormegyar.hu*, *borgyar.hu*, *szormegyar.hu* domének lesznek bejegyezve.)

1. A feladat felmérése - skálázhatóság, alternatívák, hardver.

Fontos kérdés esetünkben az, hogy a Web-szerverünk mekkora forgalmat fog lebonyolítani. Ennek mértékét találat/percben is megadhatjuk. Természetesen ez a különböző napszakokban eléggé változó lehet. Lényeges tehát, hogy ha egy egyszerű információs oldalról van szó, nem kell egy erőgépet vennünk. Ha már elektronikus áruházat is üzemeltetünk nagy számú klienssel, megfontolható nagyobb, esetleg nem PC architektúrájú gép vásárlása. A mi esetünkben talán még a cégnél meglévő egyik *Pentium*-os gép is megteszi. Minimális konfigurációnak ajánlott egy *Pentium* 166, 32MB RAM, 2 GB HD paraméterű gép. Nagyobb forgalom és nagyobb Web-hely esetén egy *Celeron* 400-as 128MB RAM és 6 GB lemez is megteszi. Extrém nagy forgalom (és CPU terhelés) esetén válasszunk nagyobb, esetleg duál-processzoros hardvert. Nem hiszem, hogy sok cég megengedhetné magának nem-x86 architektúrájú gépek beszerzését – bár azok véleményem szerint sokkal jobb hardverek, csak elterjedésüket gátolja magas áruk.

Fontos, hogy a hálózati kártya jó minőségű (pl. PCI-os 10/100 Mb/sec-es *Ethernet*) legyen, mert ez köt össze a *router*-el / tűzfallal, ez vezet a külvilágba. Természetesen fontos kérdés a sávszélesség a külvilág felé. Ez alapesetben egy 64/128k-s ISDN vonal is lehet, nagyobb forgalom esetén pl. egy 1Mb-es bérelt vonal.

Amit fontos: a hardver egységek Linux-kompatibilisek legyenek. Linux-kompatibilis hardverek: <http://lhd.datapower.com>, <http://www.linuxhardware.net>. Olvassuk el a *Linux Hardware Compatibility HOWTO*-t.: <http://www.linuxdoc.org/HOWTO/Hardware-HOWTO.html>. Főleg az alaplapon ne spóroljunk, legyen egy minőségi IDE vezérlő *chipset* rajta (pl. i440bx). Ha SCSI kártyát és merevlemezt veszünk, akkor javasolt pl. az *Adaptec* cég PCI-os SCSI kártyáit választani. Egyszerű információforrás maga a kernel: egy `make *config-al`⁸⁸ egy teljes körű listát kapunk a Linux által támogatott hardverektől. Továbbá a kernelforrás *Documentation* könyvtárában is megfelelő

⁸⁸ Bővebben a IV. Megvalósítás / 2. Finomítás / 2.4 Személyre szabott kernel konfigurálása és fordítása kézzel és a „kernel-package” csomaggal. A „lilo” beállításai. c. fejezetben.

információkhoz lehet jutni. Tudni kell, hogy a kernelben nem egy adott cég adott terméke, hanem általában annak a vezérlő-chip-je van felsorolva (leprogramozva), hiszen több termék is használhatja ugyanazt a vezérlőt. Ezért ne ijedjünk meg, hanem olvassuk át a hardver dokumentációkat, hogy melyik eszköz milyen vezérlővel rendelkezik.

Ha a cégnél jelenleg is van egy erre a feladatra kijelölhető szabad gép, akkor már csak a szoftvert és az Internet-kapcsolatot kell beszerezni. Ha nincs, akkor keressünk fel néhány számítástechnikai üzletet és szerezzük be a hardveregységeket, majd szereljük azokat össze. Ma már egy PC összeszerelése gyerekjáték, ha megfelelően választottuk meg az összetevőket. Erre itt nem térek ki, de ajánlom a következő szakirodalmat: [6].

A lényeg az, hogy mielőtt nekikezdenénk installálni a rendszert, tájékozódjunk részletesen a hardverünk Linux-kompatibilitásáról, hogy ne érjen munka közben meglepetés.

A mintapéldámban egy fiktív Bőr és Szőrmegyártó Kft. esetét vizsgálom. A menedzsment úgy határozott, hogy belépnek az elektronikus kereskedelembe, és első lépésként információkat közölnek termékeikről, szolgáltatásaikról több nyelven is az Interneten. Második lépésben pedig esetleg Online áruházat nyitnak a Web-helyükön (ezt a lépést nem tárgyalom). Mivel nincs még tapasztalatuk e téren, ezért először nem szeretnének sok pénzt fektetni a dologba. Ekkor jön a viszonylag kis teljesítményű kis költségű házi Linux-os rendszer a számításba. Felkérlik a rendszergazdát, hogy szerezzon be információkat és tervezze meg a rendszert. A rendszergazda kiválasztja és összerakja a hardvert, beszerzi a szoftvert kompakt lemezeken. (Pl. kiírja CD-re egy ismerőseivel, aki egy Internet-szolgáltatónál dolgozik, és le tudja tölteni azokat az ftp tükrökről.) Továbbá megegyezik egy Internet-szolgáltatóval az előfizetésről is.

2. Költségek becslése mintapélda alapján.

Erről szintén nehéz általánosságban nyilatkozni. Mint már említettem a szoftver beszerzési és használati költsége 0 Ft. A nagy ellenérv a TCO szokott lenni a szabadszoftverek ellen. Vagyis a termék egész használati élettartama alatti összes ráfordítás. Pl. az legyen a rendszergazda továbbképzése, az írott dokumentáció beszerzése.

Természetesen az ingyenes szoftverekhez terméktámogatás nem jár ingyen. Amennyiben szükségünk lenne telefonos vagy e-mail-es terméktámogatásra, akkor arra több cégnél is előfizethetünk. Pl. a <http://www.linuxcare.com> egy olyan cég, amely támogatást nyújt a különböző Linux rendszerekhez (Persze főleg angolul, magyarul nem). A kereskedelmi terjesztések többféle konstrukciót is nyújtanak, ha a dobozos

(vagyis több ezer forintos) termékeiket vásároljuk meg. Ennek ellenére én ezt semmiképp se javaslom, hiszen ekkor épp az ingyenességről mondunk le.

Amint a licenszekből is kiolvashatjuk, a szoftverekhez semmilyen garancia nem jár, tehát nem vonható felelősségre senki, (csak a rendszergazda) ha valami nem úgy működik, ahogy azt szeretnénk.

A legköltséghátérőbb megoldás csakis az lehet, ha a rendszergazda több Linux-os levelező listát is olvas, és oda intézi kérdéseit. Sok „borzasztó” problémáról gyorsan kiderül, hogy sokan már szembesültek ilyen szituációval, és már kész megoldással tudnak nekünk szolgálni. Ezekről az önkéntesektől ne várjunk lehetetlent, ne követelőzzünk megoldásért, mert ezek az emberek csupán jóindulatból segítenek és azért, mert emlékeznek, hogy amikor ők kezdték, akkor nekik is segített valaki. Legyünk tehát kultúráltak és ne zaklassuk kis butaságainkkal állandóan csak egy személyt, hanem minél több emberrel ismerkedjünk meg. Hiszen 1-2 esetben mindenki szívesen segít, de már a 36. levél után úgy érezheti magát az illető, hogy rászálltak.

A rendszergazdának továbbá kötelessége követni a friss Linux-os híreket. Pl. <http://www.linux.hu>, <http://slashdot.org>, <http://www.linux.org>, <http://www.linux.com>, <http://freshmeat.net>, <http://linuxapps.com>, stb. Figyeljük a biztonsággal foglalkozó helyeket, levelezőlistákat is, pl. security-l@sunserv.kfki.hu

Ha a rendszer mindig tartalmazza a biztonsági javításokat, és rendszeren be van konfigurálva, akkor szinte rá se kell néznünk.

Az üzemeltetési költséghez tartozik a leállítás is, hiszen a rendszergazdát elő kell keríteni, hogy indítsa újra a gépet és keresse meg a hiba okát.

„A költséghatékonysághoz tartozik a kiesett állásidő kérdése is. Linuxos gépek esetében ez minimálisra csökken, nem ritka az sem, ha egy gép 100-200 napot üzemel leállítás nélkül. Például Donald Becker „A szövegszerkesztőtől a szuperszámítógépig” címmel tartott előadást a jelenlegi Beowulf projektjéről, és beszélt a rendszerek stabilitásáról is. A legtöbb rendszere már 100 napja folyamatosan működik, de van olyan is, amelyhez 200 napja nem kellett hozzányúlni. Ez nem attól érdekes, hogy egy gép ennyi ideig bírja, hanem, hogy nagy számú (100-200) gép működik együtt ennyi ideje.”⁸⁹

Véleményem szerint a TCO minimalizálható, ha a tervezéskor betartunk minden szabályt, és nem saját magunk ellen dolgozunk. A megvalósításnál természetesen nem szabad nagy kompromisszumokat kötni a terv ellenében, hiszen akkor felesleges volt a tervezési fázis. A „Jó lesz az, hidd el!” alapon végzett munka mindig félmunka.

⁸⁹ <http://www.vmg.sulinet.hu/vmg/home/szamtech/linux/node23.htm> Egyébként olyan gépről is olvastam, mely 435 napja meggy folyamatosan.

Ha egy már meglévő gépre telepítjük a rendszert, akkor a hardver költsége is minimális lehet (egyéb kiegészítőkre mindig szükség lesz). Ha rászánunk egy kis pénzt, akkor igénytől függően összeállíthatunk pl. egy *Celeron*-os gépet 100-150 ezer Ft + ÁFÁ-ból. Egy komolyabb duál-PII-es gépet is ki lehet talán hozni 300 ezer Ft-ból. A fontos az, hogy viszonylag minőségi és elterjedt alkatrészekből építkezzünk, mert így sok fejfájástól kímélhetjük meg magunkat. Termékneveket nem akarok megemlíteni, többek között védjegyi okok miatt is – egyébként mindenkinek a maga ízlése szerint. Mindenki attól a cégtől vásárol, amely termékeivel jó tapasztalatai vannak. Sokat lehet vitatkozni, hogy ki mikor milyen egységgel hogyan járt, miközben a másik termék kitűnően működött, *vica versa*.

Konkrét árakat pedig azért is badarság lenne említenem, mert az már a jövő héten nem lenne igaz, nemhogy mikorra az olvasó ezt a könyvet a kezébe veszi.

3. Biztonság

Megfelelő rendszertervezéssel elejét lehet venni a legkülönbébb támadásoknak. Ha betartjuk a *II. Alapfogalmak / 8. Biztonsági alapok (hardver, szoftver)*, *IV. Megvalósítás / 2. Finomítás* fejezetekben leírt szabályokat, továbbá a következő tanácsokat is betartjuk és alkalmazzuk, akkor nagymértékben fokozhatjuk a rendszerünk betörés-biztosságát és helyes működését.

3.1 Milyen programok lehetnek / nem lehetnek egy Web-szerveren?

Egyes programokat / szolgáltatásokat egyáltalán nem tanácsos és felesleges egy éles Web-szerveren futtatni és / vagy tárolni, mert támadási felületet biztosíthatnak a behatolók számára. Bár egyazon szerveren rengeteg szolgáltatást tud nyújtani a Linux, igazából, biztonsági okokból egy külön erre a célra kijelölt gépet kell Web-szerverként üzemeltetnünk. A „mindent egybe” filozófiát pedig felejtsük el. Néhány fontos tanács:

- Ne legyen semmilyen fordító és fejlesztő eszköz / program. Ezek lehetőséget adnak a betörőnek, hogy trójai faló programokat fordítsanak a rendszeren.
- Ne legyenek *NFS export*-ok, NFS szerver. Az NFS-t *No File Security*-nek is szokták csúfolni. Rajta keresztül könnyen feltörhető a rendszer.
- Ne legyen NIS.⁹⁰
- Ha lehet ne legyen FTP szerver. Helyettesítsük *scp*-vel. Ha viszont lesz, akkor ne legyen *anonymous ftp* szolgáltatás. A felhasználók legyenek *chroot*-olva a saját *home*-könyvtárukba.
- Ne használjunk *telnet* szolgáltatást. Helyettesítsük pl. *ssh*-val. (lásd később)
- Ne fussanak az *r** és RPC szolgáltatások (lásd később)
- Ne fusson semmiféle felesleges szerver, pl. *ircd*, *talkd*. Továbbá ne tartsunk veszélyes klienseket (*irc*, *icq*, stb.)

⁹⁰ Network Information System: a Sun cég egy régebbi, nem biztonságos megoldása a felhasználók azonosítására gépek között. A felhasználónak Csak egyszer kell bejelentkeznie a hálózatba, ezután a gépek a NIS segítségével azonosítják azt egymás között.

- Ne szolgáltatassunk ki a felhasználókról információkat (`fingerd`, `identd`)
- A különlegesen fontos fájlknál állítsuk be az *immutable bit*-et.⁹¹
- Szigorú `umask`⁹² érték elhelyezése az `/etc/profile`-ban, és a felhasználók egyéni beállításaiiban.
- A rendszergazda számára elhelyezhetünk 077-es `umask`-ot is, de ezt csak a rendszer készre-tétele után érdemes. Ekkor a jog a következő lesz: `-rw-----`
- Lehetőség szerint csak eredeti Debian csomagokat használjunk, ha viszont fordítanunk kell, próbáljuk meg először a forráskódot a Debian tükörről leszedni. Ha onnan nem tudjuk, akkor a készítő hivatalos honlapjáról, vagy ftp helyéről töltsük le, esetleg a hivatalos magyar tükörről. Erre azért van szükség, mert egy ismeretlen helyről beszerzett bináris program vagy forráskód tartalmazhat kiskapukat – tehát lehet, hogy kompromittált. A kernel és a Debian csomagok és források MD5-ös igazolással érkeznek, mely alapján ellenőrizni lehet a fájl integritását. A fordítást mindig másik gépen végezzük.

Karbantartás során:

- Rendszeresen ellenőrizzük, hogy mely programok rendelkeznek SUID és vagy SGID bit-ekkel. Ezt megtehetjük a következő paranccsal:
`find / -type f \( -perm -04000 -o -perm -02000 \)`
- Rendszeresen ellenőrizzük, hogy mely fájlok írhatóak mindenki által. Ezt megtehetjük a következő paranccsal:
`find / -perm -2 ! -type l -ls`
- Rendszeresen ellenőrizzük, hogy mely fájloknak nincsen tulajdonosuk. Ez azokra a fájlokra is jellemző lehet, melyek betörés céljára vannak használva. Ezt megtehetjük a következő paranccsal:
`find / -nouser -o -nogroup -print`
- Keressük meg az `.rhosts` fájlokat. Ezek betöréshez könnyen felhasználhatóak ezért töröljük őket, ha nincs rájuk különösen indokolt szükség.
`find /home -name .rhosts -print`

Ezeket a fenti kereséseket betéve egy shell-script-be és azt a `cron` időzítő segítségével mindennap lefuttatva, ennek kimenetét a `/var/log`-ban elhelyezve, vagy e-mailben elküldve automatizálhatjuk. Ezek a lépések elhagyhatóak, ha egy jól beállított `tripwire` programot működtetünk.

⁹¹ Lásd: `man chattr(1)`: „Egy 'i' attribútumú fájlt nem lehet módosítani. Nem lehet törölni, átnevezni, hozzáfűzni, benne adatot átírni és semmilyen kötést (link) rá létrehozni. Csak `superuser` (`root`) tud adni vagy elvenni ilyen attribútumot.” Magyar fordítás: Németh Péter ggenpete@gold.uni-miskolc.hu

⁹² E program segítségével azt állíthatjuk be, hogy az újonnan keletkező fájlok / könyvtárak milyen kezdő jogosultsággal rendelkezzenek. Az alapbeállítás 022. Ekkor `rw-r--r--` lesz a jog. Állítsuk át legalább 027-re. Ekkor `-rw-r----` lesz a jog.

3.2 A partíciók megtervezése általánosan és a mintapéldához

Fontos egy éles rendszer esetében szétválasztani a különböző funkciót betöltő alkönyvtárakat különböző partíciókra és egyeseket csak olvasható módban használni. Ez nagyban növelheti a rendszer hitelességét⁹³, és a mentéseket is leegyszerűsítheti.

A mai kernelek már támogatják (a 2.2-es folttal, a 2.4 alapból) az *LVM (Logical Volume Management, Logikai Kötetkezelés)*-t. Ezzel a módszerrel különböző merevlemezekről vagy egyazon lemezről, de különböző helyekről fűzhetünk össze partíciókat egyetlen egységgé, melynek mérete ezért dinamikusan változtatható. Itt erre a módszerre nem térek ki, csupán a hagyományos utat tárgyalom. Az *LVM*-et csak haladóknak ajánlom.

A következő táblázat tartalmaz egy lehetséges kiosztást. Esetünkben egy egyszerű IDE vezérlős ATA merevlemezről van szó, mely az első (*ide0*) vezérlő „*master*” részére van csatlakoztatva. Ezeket az opciókat csak a rendszer készre-tétele után szabad beállítani. Ha be lett állítva, teszteljük a rendszer újraindítását, hogy mennyire fog zökkenőmentesen zajlani.

csatolási pont	méret kb.	mount opciók	eszköz
/	40-60 MB	ro ⁹⁴ ,defaults ⁹⁵	hda3
/boot	10 MB	ro,nosuid,noexec ⁹⁶ ,nodev,defaults	hda1
/usr	tetszőleges	ro,nodev,defaults	hda5
/home	tetszőleges	nosuid,noexec,nodev,defaults,usrquota	hda6
/tmp	100 MB	nosuid,noexec,nodev,defaults,usrquota	hda7
/var	tetszőleges	nosuid,noexec,nodev,defaults	hda8
/var/www	tetszőleges	nosuid,noexec ⁹⁷ ,nodev,defaults	hda9
swap	mem x 2	(ezt a rendszer nem fűzi fel)	hda2

5. táblázat - Partíció-terv

A rendszer indulásakor az inicializáló script automatikusan felfűzeti a *mount* programmal a */etc/fstab* fájlban szereplő bejegyzéseket. Ez a program az adott partícióra vonatkozó paramétereket is az *fstab*-ból olvassa ki. Az opciók jelentése:⁹⁸

„A következő opciókat minden fájlrendszer esetén alkalmazhatjuk:

- ***async*** A fájlrendszer minden írási/olvasási művelete aszinkron módon megy végbe.⁹⁹

⁹³ Itt bővebb információk találhatóak a particionálás technikai oldaláról:

<http://www.vmg.sulinet.hu/vmg/home/szamtech/linux/node94.htm>

⁹⁴ Vigyázat! Ez esetben nem működhet pl. a *syslog-ng* és a *MAKEDEV*. (Bővebben később)

⁹⁵ Ahhoz, hogy *ro*-ban lehessen használni ki kell küszöbölni azt, hogy *ide* írnia kelljen a rendszernek. (Bővebben később)

⁹⁶ Ez sajnos könnyen kijátszható.

⁹⁷ Vigyázat! Ezzel megakadályozhatjuk a CGI scriptek futtatását!

⁹⁸ Ezt a felsorolást a „*man mount*”, vagyis a *mount* manual page-éből vettem át. Magyar fordítás: Horváth András horvatha@rs1.szif.hu

- **atime** Frissíti az inode¹⁰⁰-ok elérési ideit minden elérés esetén. (Alapértelmezett.)
- **auto** A fájlrendszer csatolható a -a opcióval.¹⁰¹
- **defaults** Az alapértelmezett opciókat használja. Ezek: rw, suid, dev, exec, auto, nouser, és async.
- **dev** Értelmezi a karakteres vagy blokkos speciális eszközfájlokat a fájlrendszeren.
- **exec** Megengedi a bináris fájlok futtatását.
- **noatime** Nem frissíti az inode-ok elérési ideit minden elérés esetén. (Ez hasznos lehet pl. a 'news spool' gyorsabb elérésének biztosítására news szerverek esetén.)
- **noauto** Csak kifejezett parancsra csatolható, azaz pl. -a opcióval nem csatolódik.
- **nodev** Nem értelmezi a karakteres vagy blokkos speciális eszközfájlokat a fájlrendszeren.
- **noexec** Nem engedi meg a csatolt rendszeren található bináris fájlok futtatását. Ez pl. akkor hasznos, ha egy szerver más architektúrájú binárisokat is tartalmazó fájlrendszert használ, mint a sajátja.¹⁰²
- **nosuid** Nem engedélyezi a set-user-identifier (suid) és set-group-identifier (sgid) bitek használatát.¹⁰³
- **nouser** Megtiltja minden közönséges (nem root) felhasználónak a fájlrendszer csatolását. (Alapértelmezett.)
- **remount** Megkísérli egy már csatolt fájlrendszer újbóli csatolását. Ezt arra szokás használni, hogy más opciókkal csatoljuk újra a fájlrendszert, például egy korábban csak olvasható fájlrendszert írhatóvá tegyünk.
- **ro** Csak olvasható módon csatolja a fájlrendszert.
- **rw** Írható/olvasható módon csatolja a fájlrendszert.
- **suid** Engedélyezi a set-user-identifier (suid) és set-group-identifier (sgid) bitek használatát.
- **sync** A fájlrendszer írási és olvasási műveleteit szinkronizáltan végzi.
- **user** Megengedi minden közönséges (nem root) felhasználónak a fájlrendszer csatolását. Ez az opció bekapcsolja a noexec, nosuid, és nodev opciókat, hacsak nem a további opciók ezt felülbírálják. (Biztonsági okokból ezt csak nagyon átgondolt esetekben szabad megtenni.)”

Tehát azok az alkönyvtár-rendszerek, melyek elméletileg nem tartalmazhatnak futtatható fájlokat (/var /tmp /home /boot) ne is legyenek képesek futtatásra. Továbbá a megfelelő partíciókon ne lehessen eszközfájlokat létrehozni, és az eszközökhöz hozzáférni, továbbá ne lehessen tulajdonos-váltó programokat se indítani. Azok a partíciók, melyek nem szabad, hogy változzanak (csak a rendszergazda változtathatja meg őket), csak olvasható formában legyenek felfűzve. Egy további trükk lehet az is, ha a ro-nak szánt partíciókat egy iso9660 (CD image)

⁹⁹ async mód esetében memória gyorsítótár működik, sync esetében direktben a lemezre ír. Az async nagyobb teljesítményt, a sync nagyobb biztonságot nyújt.

¹⁰⁰ inode: a Linux-os ext2 fájlrendszer alapegysége

¹⁰¹ vagyis az indulási folyamatkor is.

¹⁰² Sajnos ez a védelem könnyen **kijátszható**: /lib/ld-linux.so fájlnev – és elindul.

¹⁰³ vagyis nem válthat a program másik felhasználói jogkörre indulás után.

formában készítjük el. Ekkor az esetleges betörő hiába szerez rendszergazdai jogköröket, nem fogja tudni újra felfűzetni ezeket a fájlrendszereket *rw*-ben, mert az *isofs* maga csak olvasható. Ez persze további bonyodalmakkal jár és kissé nehézkes a rendszer frissítése, lévén, hogy mindig egy új ISO-fájlt kell készíteni.

1. A */boot* partíciót azért tesszük lefelőre és külön, mert:

- Így biztosan látni fogja a *lilo*¹⁰⁴.
- Jobb, ha *ro* és senki nem nyúl bele, ugyanis a kernel sérülékeny pontja a rendszernek.

2. A második a **swap** partíció, mely virtuális memóriát képez a lemezen. Ezzel a fizikai RAM 2-3 szorosáig tudjuk optimálisan bővíteni a rendszert. Ha pl. 64 MB RAM van a gépben, akkor ajánlott 128 MB *swap*-ot alkalmazni. Természetesen terheléstől függően lehet, hogy egyáltalán nem, vagy csak kis mértékben lesz használva a *swap*. Ha 128 MB memóriánk van, nem biztos, hogy kell 256 MB *swap*. Teszteljük a rendszert. Egy *swap* partíció maximális mérete 2 GB, de lehet több ilyen partíció is. Akár több lemezen is. Azért érdemes előre tenni a *swap* partíciót, mert ha azt a rendszer sokat használja, akkor így jelentős sebesség növekedés érhető el azzal szemben, mint amikor az a lemez „végén” helyezkedik el.

3. A harmadik a **gyökér** fájlrendszer, mely a */bin*, */lib*, */sbin*, */etc*, */root*, */dev*, */mnt* könyvtárakat tartalmazza. Ezek olyan fájlokat tartalmaznak, melyeket csak a rendszergazdának javasolt megváltoztatni. Ezért ha már működik a rendszer és mindent jól beállítottunk és leteszteltünk, tegyük írásvédetté a gyökér partíciót. Ahhoz, hogy ez problémamentesen sikerüljön, olvassuk el a *IV. Megvalósítás / 2. Finomítás / 2.13 A „/etc/fstab” és az „init script”-ek beállítása* című fejezetet.

4. A negyedik partíció egy „**extended**” vagyis kibővített partíció. Ez azért nincs feltüntetve a listán, mert ez tartalmazza a többi logikai partíciót. A logikai partíciók számozása mindig öttől kezdődik, akár van 2,3,4-es elsődleges partíció, akár nincs.

5. Az ötödik partíció az első logikai. Ide a */usr* alkönyvtár tartozik, melyre a rendszer indításához és alapvető működéséhez nem szükséges programokat teszi a telepítő. Mivel egy rendszerbeállítás után ennek sem szabad változnia, ezért ezt is *ro*-ban használjuk. Ez mindaddig kis méretet igényel, amíg kevés funkciót teljesít a szerver. Ekkor akár 50MB-al is beéri. Ha sokfunkciós alkalmazás-szervert építünk, akkor felmehet akár 1-2GB-ra is a mérete.

¹⁰⁴ Bővebben: *IV. Megvalósítás / 1.2.2 Szükséges alapbeállítások, partícionálás*

6. A hatodik tartalmazza a `/home` könyvtárat, mely a valódi felhasználók könyvtárait és fájljait tárolja. Mivel ebben a rendszerben nem lesz sok felhasználó – hagyományos értelemben jobbra egy se – ezért ez lehet elég kicsi is. Igény szerint állítsuk be a méretét. A felhasználóknak biztonsági okokból ne engedélyezzük indítható fájlok futtatását, eszközfájlok létrehozását és *setuid*-al való kísérletezgetést se. A mérete az adott rendszertől, vagyis a felhasználók számától függ. Általában legyen minél nagyobb, ha sok a felhasználó és minél kisebb, ha nincsenek valódi felhasználók, csak rendszergazdák. Pl. legyen 500 MB.
7. A hetedik az **átmeneti fájlokat** tartalmazó partíció. Ennek a könyvtárnak ún. „sticky” vagyis kb. „ragadós” bit-je van. Ez azt jelenti, hogy az adott folyamat kapja meg a saját maga által létrehozott fájl a tulajdonjogát. Pl. ha egy `lali` nevű felhasználó által futtatott program létre hoz egy átmeneti fájlt, akkor annak a tulajdonosa a `lali` lesz. Ekkor ezzel a fájllal azt tesz, amit akar, de a többiek fájljait nem tudja piszkálni, esetleg olvasni se – ha megfelelő az *umask* értéke. Sok felhasználó és egyszerre futó folyamat esetén kevés lehet a 100 MB, ekkor növeljük meg igény szerint. Ha olyan programokat használunk, melyek extrém nagy fájlokkal dolgoznak (hang, videó), akkor elég hamar elfogyhat ez a hely.
8. A nyolcadik a `/var` könyvtár: ez alatt találhatóak az állandóan változó adatok, mint pl. a levelezés, a rendszernapló, a csomag-adatok, stb. Továbbá itt lesznek elhelyezve a `mysql` adatbázisok is, ezért kellő mennyiségű helyet kell biztosítani ezek számára. Ne felejtsük el később a `/var/lib/mysql` könyvtárt naponta menteni.
9. Itt van a Web-szerver szíve-lelke, a Web-tartalom. Ez a `/var/www` könyvtár a Web-szerver dokumentum-gyökere. Ennek mérete a Web-hely mérete szerint kell, hogy alakuljon.

3.3 A biztonsági mentés (backup) lehetőségei, módszerei és javasolt paraméterei

Ez szintén egy ritkán betartott szabály és fontos kérdés. Jó mentés nélkül a rendszer nem sokat ér, nem biztonságos. A mentési „politikát” szabályzatban kell rögzíteni, be kell tartani és tartatni az eredményes helyreállítás érdekében. A következőket kell meghatározni: „

- *milyen célra*
- *milyen gyakorisággal*
- *milyen eszközzel*
- *mely fájlrendszereket kell lementeni*

A mentés céljai lehetnek:

- *Hardver – főként merevlemez – hiba miatti teljes összeomlás esetén a rendszer gyors visszatöltését biztosítani lehessen.*
- *Biztonsági tartalék – egy tiszta, idegen behatolók aknáitól mentes rendszer – gyanított, vagy tényleges betörés esetére.*
- *A betörés és károkozás történetének utólagos felderítéséhez.*
- *Felhasználók állományai, arra az esetre, ha letörlik, elvesztik, rosszul javítanak bele.*
- *Számlázási vagy bármilyen jogi vita esetére bizonyíték. ”[12 p. 61]*

Mentési eljárás típusok:

- Telepítés utáni mentés: ekkor még semmilyen felhasználói adat nincs fenn.
- Teljes mentés: az összes partíció mentése.
- Inkrementális (réteges) mentés. Csak az utolsó mentés után megváltozott fájlok kerülnek mentésre.

Azokat a partíciókat, melyek csak olvasható formában vannak, elég egyszer elmenteni (vagyis csak minden változtatás / frissítés után). A folyamatosan változó partíciókat gyors változások esetén érdemes akár naponta is elmenteni.

A mentéssel szembeni követelmények:

- Sokáig őrizze meg a média az adatot, nagyon hibatűrő média kell.
- Szabványos, kompatibilis, sokáig fennmaradó technológia.
- Többször felhasználható, nagy kapacitású, tartós média
- Címken fel legyen tüntetve a mentés dátuma, a gép neve, a mentési szint, ha több mentés is ráfér, akkor mind.

A média fizikai védelme: Mivel a mentés tartalmazza az összes bizalmas adatunkat is, ezért lehetőleg páncélszekrényben kell tárolni, hőtől, fénytől, víztől, párától, mágneses és elektromos tértől elzárva. Védeni kell lopás ellen (és) szállítás közben is.

Legegyszerűbb az, ha a gépben van egy **CD-író**, benne egy újraírható kompakt lemez és a mentés minden éjszaka megtörténik a rendszer időzítő naplójából (/etc/crontab). Ennek a megoldásnak a hátránya az, hogy ha teljes mentést kell végezni, akkor kicsi lehet a 650Mb-os tárterület, továbbá a CD-k cseréje nélkül mindig csak egy mentésünk lehet, ami nem tanácsos. Továbbá szükség van egy ugyanekkora átmeneti tárterületre az ISO image fájl elkészítéséhez, mely az írás előtt szükséges. Ez a megoldás viszonylag gyorsnak, és tartósnak mondható. Ma már költséghatékony is. A másik egyszerű megoldás egy nagyteljesítményű és tárolókapacitású **szalagos egység**. Ezekből elég nagy a kínálat. 2GB-os méret alatt nem ajánlom, hogy egységet vásároljunk. Javasolt akkora kapacitású egységet vásárolni, hogy egy teljes mentés egyben ráférjen. Linux alatt elég jól használhatóak az *Ftape* jellegű meghajtók, melyek az alaplap floppy vezérlőre csatlakoztathatóak. A szalagos egységek közül vannak párhuzamos portra kapcsolhatóak és SCSI-s felületűek is. Az utóbbiak elég drágák is,

ui. a SCSI vezérlőt is meg kell venni. Cserébe viszont nagyobb biztonságot kapunk. Valamint nagyobb sebességet és kapacitást is.¹⁰⁵

A fájlokat általában tömörítve érdemes menteni. Vannak olyan szalagos egységek, melyek hardveres tömörítést alkalmaznak. A CD-RW esetében azonban nekünk kell gondoskodni szoftveres tömörítésről is. Így a kapacitás akár kétszeres is lehet.

A következő fő kérdés a mentés szoftverének kiválasztása. Vannak a rendszerhez járó általános archiváló programok, mint amilyen a `tar` és a `cpio` is. A `dump` program azonban fájlrendszer-szintű mentést készít.

„*A dump abban különbözik ezektől¹⁰⁶, hogy a fájlrendszer tartalmát közvetlenül, nem a fájlrendszeren keresztül olvassa. Ezt speciálisan biztonsági mentések céljából írták, míg a tar és cpio programokat elsősorban archiválásra, de azért használhatók biztonsági mentésre is.*

A fájlrendszer közvetlen olvasásának vannak előnyei. Lehetséges ilyenkor a fájlok visszaállítása időbélyegjeik átállítása nélkül; a tar és a cpio használata előtt viszont a fájlrendszert először csak olvashatóan kell csatlakoztatni (mount-olni). A fájlrendszer közvetlen olvasása hatékonyabb is, ha mindent le kell menteni, mert a legkevesebb fejmozgással megoldható. A legnagyobb hátránya az, hogy a mentési program ilyenkor a fájlrendszer típusához kötődik: pl. a Linux dump utasítása csak az ext2 fájlrendszerre működik. A dump továbbá közvetlenül támogatja a mentési szinteket míg a tar és a cpio esetén ezt egyéb eszközökkel kell megvalósítani.”¹⁰⁷

Természetesen vannak a „fapados” `dump` helyett más mentési programok és eljárások is. Ezek egy része viszont kereskedelmi termék. A `dump`-nál a hangsúly az automatizálhatóságon és a távoli gépekre történő mentésen van és nem a színes grafikus felületen. Főleg szalagos egységekhez használható. Segítségért forduljunk a manuálhoz.

A *Potato*-ban több professzionális mentési programrendszer is található. Ezek főleg központi *backup*-szerverre dolgoznak és kliens-szerver rendszerűek. Az egyik ilyen program az *afbackup*. Nagy hangsúlyt helyeznek a biztonságra és a kliens azonosítására. A *backup*-szerverről is indítható a kliensről való mentés. Lehetőség van a tömörítésre, és a partíciók közvetlen mentésére is.

Először is olvassuk el a dokumentációt. Ez segít a rendszer megértésében. Ha nincs külön gépünk a mentésre, de a szervernek van szalagos egysége, akkor sincs baj. A mentő program szerver része foglalkozik a szalaggal, a kliens része pedig a fájlokkal. A

¹⁰⁵ Böszöri barátom megjegyzése: „Egy újabb (HP gyártmányú) DAT, vagy (Exabyte) 8mm-es szalagos egység több GB-ot is felír óránként, egy floppy csatlós streamer ezt nem éri utól se sebességben, se kapacitásban. Ha olcsóbb megoldást keresünk, akkor egy IDE felületű Travan szabványú egység is jó lehet, ennek a kapacitása elérheti a (középkategóriás) DAT szalagét (3.5 - 8 GB), viszont sebességben alulmarad. Az ide-tape modul kezeli.” zboszor@externet.hu

¹⁰⁶ értsd: `tar/cpio`

¹⁰⁷ <http://www.vmg.sulinet.hu/vmghome/szamtech/linux/node313.htm>

program beállítása igen egyszerű: elvégezhető az `/etc/afbackup` alatt lévő fájlokon és egy automata *script* program segítségével is.

Mivel a mentés mindenkinél más típusú egységre történhet, ezért eltekintek a konfigurációs fájlok részletes bemutatásától.

Egy másik hasznos mentő program a Potato-ban az **amanda**. Ezt inkább csak külön szalag-szerverekhez ajánlják. Továbbá ott van még a **kbackup**, **taper**, **tob** melyeket főleg az egygépes mentésre terveztek. Mindenkinek ajánlom, hogy próbáljon ki többet is, majd az igényeinek megfelelőt válassza. Én a mintapéldában a **kbackup**-ot használom.

A CD-írók mentéshez a következő programokat ajánlom:

cdbbackup: <http://cdbbackup.home.dhs.org>

scdbbackup: <http://scdbbackup.linuxbox.com>

Továbbá szükségünk lehet a CD írásához a **cdrecord** és az **mkisofs** programokra.

Egy *backup* programokat összefoglaló hely:

http://linuxberg.externet.hu/conhtml/adm_backup.html

3.4 Szoftveres UPS (szünetmentes tápegység) felügyelet soros porton keresztül

Ma már elképzelhetetlen egy szerver szünetmentes tápegység (*UPS*, *Uninterruptible Power Supply*) nélkül. Az újabb típusok alkalmasak kapcsolatot teremteni a szerverrel és hosszabb áramszünet esetén megkérni azt, hogy álljon le. Ez általában a szerver soros portján keresztül történik.

Fontos paraméterek:

- van-e a tápegységnek soros vagy más kapcsolati lehetősége a szerverhez
- támogatja-e a Debian-ban lévő UPS felügyeleti démonok egyike a tápot¹⁰⁸
- mekkora az áthidalási idő, a teljesítmény
- mekkora az akkumulátor újratöltődési ideje
- mennyi az akkumulátor élettartama
- milyen hosszú a garancia
- kapható-e alkatrész, új akkumulátor a táphoz
- van-e túláram és/vagy túlfeszültség-védelem
- mennyibe kerül

A lehetőségek szerint válasszunk olyat, aminek rövid az újratöltődési ideje, van benne védelem, és persze kompatibilis a szoftverünkkel. Ne sajnáljuk rá a pénzt, ez egy fontos alkatrésze rendszerünknek. Olyan 40-50 ezer forintból már jó készüléket lehet venni. Ha a szerverünk a szolgáltatónál lesz, lehet, hogy ott biztosítanak neki

tápegységet, bár annak hátránya az lehet, hogy nem kapcsolja le automatikusan a szerverünket.

A Debian-ban a következő szoftverek találhatók:

- **apcd**: Az APC Smart UPS termékeihez.
- **apcupsd**: Az APC cég termékeihez készült, a táp jelzése esetén kikapcsolási folyamatot indít. Támogatja a *BackUPS*, *BackUPS Pro*, *SmartUPS V/S*, és *SmartUPS(NET/RM)* termékeket.
- **bpowerd**: A *Best Patriot* cég termékeit támogatja.
- **genpower**: a legtöbb RS-232-es eszközzel működik. Extra képességei: hálózati feszültség érzékelése, lemerült akku felismerése, soros kábel felismerése, az UPS inverterének kiiktatása. A Debian csomagban van egy extra kábelhez való vezérlő is, melyet „*apc-pnp*”-nek hívnak. Segítségével a fenti extrák kihasználhatóak a következő termékeknél: APC Back-UPS Pro, Smart-UPS, és Matrix-UPS rendszerek.
- **upsd**: képes a hálózaton keresztül is menedzselni a szerverek UPS-eit.
- **powstatd (-crypt)**: könnyen konfigurálható, főleg *dumb*¹⁰⁹ jellegű UPS-ekhez. a következő termékeket támogatja: *CyberPower PowerSL* soro, *CyberPower Power2000 1500VA*, *CyberPower Power99 325VA, 400VA, 500VA, 720VA*, Néhány régebbi *CyberPower 385VA, 450VA modell*, *TrippLite Internet Office 500 UPS*, több régebbi APC termék. Ez a program arra is képes, hogy több gépet is lekapcsoljon, melyek egy tápegységről futnak.

Nem ajánlhatok a fentiek közül „legjobb” címen programot, mert nincs lehetőségem kipróbálni az összes tápegységet. Mindenki válassza ki a neki szimpatikusát, esetleg tegyen fel többet is és próbálja ki mind. Természetesen az itt felsorolt eszközökön kívül is léteznek olyanok, melyekhez írtak Linux-os vezérlőprogramot (pl. MGE). Ezt a gyártó Web-helyén megtudhatjuk.

Fontos, hogy ha lehet, úgy állítsuk be a felügyeleti szoftvert, hogy küldjön levelet minden áramkimaradásról a rendszergazdának és legalább egy helyi dolgozónak, aki el tudja rögtön hárítani a hibát, ha leállna a szerver.

Lényeges, hogy a szerver BIOS-át ATX-es ház esetén – ha lehet – úgy állítsuk be, hogy áramkimaradás után azonnal bekapcsoljon, amint újra van áram.

¹⁰⁸ Vagy van-e Linux-os vezérlőprogramja valahol – ez az út körülményesebb lehet.

¹⁰⁹ Ez itt „butát” jelent. Tehát nem az újabb intelligens megoldásokat.

3.5 A szükséges felhasználók/csoportok és a lemezkvóta megtervezése

Mindenek előtt szükséges leszögezni, hogy csak azok a személyek jussanak felhasználói jogosultsághoz (*UNIX user account*-hoz) a szerveren, akiknek erre tényleg szüksége van. Egy felhasználói számlát többen ne használjanak.

A rendszergazdai jelszót kettőnél se több, se kevesebb ember ne tudhassa. Csak olyan ember ismerje a rendszergazdai jelszót, akiben a cégvezetés nagyon megbízik.

Mint már ahogyan a *II. Alapfogalmak / 6. A Web-személyzet felépítése* c. fejezetben felvázoltam, a gépen dolgozó emberek különböző funkciókat töltenek be, és más-más feladataik vannak. Ezért más-más hozzáférési jogosultság is szükséges számukra.

A fő rendszergazda kapja meg a *root* jogosultságot. Ezenkívül neki is létre kell hozni egy felhasználói számlát, mert úgy lesz a rendszer beállítva, hogy *root*-ként csak a konzolról lehessen bejelentkezni (vagyis pl. *ssh*-val sem). Ha már bejelentkezett a gazda, akkor – ha szüksége van rá – átléphet a *su* paranccsal *root* szerepkörbe. Ez azért is előnyös, mert így két jelszót is fel kell törni ahhoz, hogy valaki bejusson. Továbbá napi teendőjének nagy részét nem kell *root*-ként végeznie, így megkerülhet sok csapdát és véletlen törlést is az ember. Csak azokat a tevékenységeket végezzük *root*-ként, amit nem lehet *user*-ként megtenni.

A *webgazda* teszi fel és tartja karban az oldalakat. Ő a *www-data* csoport tagja. Amennyiben többen is végzik ezt a munkát, legyen e csoport tagjainak olyan *umask* érték beállítva, mely lehetővé teszi a csoport tagjai számára az írás/olvasást is. Ha beírjuk a következő parancsot: *umask -S*, akkor érthetőbb formában tudhatjuk meg a jelenleg érvényes *umask* értéket. Pl. *u=rwx,g=rx,o=rx*. Ha ezt be szeretnénk állítani *u=rwx,g=rwx,o=-ra*, akkor megtehetjük *umask -S u=rwx,g=rwx,o=-* paranccsal. Figyelem! Az Apache program *www-data* jogokkal fut. Ha egy olyan script-et írunk, vagy lehetőséget hagyunk a Web oldalon vagy programokban, akkor az Apache-nak, rajta keresztül pedig a támadónak írási joga lesz a szerver tartalmára (vagyis a Web-oldalakra.) Ezért jobb, ha egy külön csoportot hozunk létre, pl. *web-csop* néven. Ebbe a csoportba tartozzanak azok a felhasználók, akik a Web-tartalmon módosíthatnak közvetlenül a szerveren. A másik megoldás, hogy pl. a fájlok a *webgazda* tulajdonában vannak, de a *www-data* csoport a csoportazonosítójuk.

Ideális esetben nincs sok változás és elég, ha csak a *webgazda* kezeli az oldalakat. Ha viszont több ember dolgozik a szerveren, és gyorsabban változik a tartalom, pl. a fejlesztés alatt, akkor a többi embernek is lehet adni felhasználói számlát a gépre. Ezek az emberek legyenek a *web-csop* tagjai. Az általuk feltett fájlok jogosultságaival nekik kell törődni.

Ha azt szeretnénk, hogy az Apache program beállításait is a *webgazda* kezelje, akkor adjuk át neki az `/etc/apache-ssl/` könyvtárat és fájljait. Ezt megtehetjük így is:

```
chown -R webgazda:webgazda /etc/apache-ssl
```

Azt is megtehetjük, hogy a fájlok a *root* tulajdonában maradnak, de a csoportját adjuk át a *webgazdának*, majd írási jogot kap a fájlokra a csoport.

Az ideális esetben van külön Web-fejlesztői gép. Ekkor így néz ki a felhasználók listája:

felhasználó	felhasználó név	csoport tagja
rendszergazda	root, rgazda	root, rgazda (minden jog)
Web-rendszergazda	webgazda	webgazda, www-data, web-csop, users
adatbázis-rendszergazda	abgazda	abgazda, mysql, users

6. táblázat - Felhasználók listája 1.

Ha nincs külön gép, akkor a fejlesztés is a szerveren zajlik. Ekkor:

felhasználó	felhasználó név	csoport tagja
Web-grafikus	webgraf	webgraf, webcsop, users
Web-programozó	webprog	webprog, webcsop, users
Web-tervező	webterv	webterv, webcsop, users
titkárnő	titkarno	titkarno, users

7. táblázat - Felhasználók listája 2.

Igény szerint vehetünk fel más felhasználókat is, például az adatbázis feltöltéséhez és karbantartásához. Ezek a felhasználók nem hozhatnak létre új táblákat, nem törölhetnek meglévőket, nem módosíthatják az adatbázis szerkezetét, csupán feltölthetik adatokkal azt. Külön jogosultságrendszerrel megadható az is, hogy melyik táblához ki és hogyan férhet hozzá. Fontos kiemelni, hogy a MySQL saját felhasználó és jelszó adatbázissal rendelkezik. Ezeket a jogosultságokat az adatbázis-rendszergazda tudja kezelni. Ahhoz, hogy valakinek hozzáférése legyen az adatbázishoz, nem kell, hogy UN*X számlája legyen a gépen. Erre csak akkor van szükség, ha az illetőnek be kell jelentkeznie a gépre, hogy egy adatbázis klienst használjon. Mivel a UN*X-os kliens eléggé fapados az átlagfelhasználónak, ezért kétlem, hogy erre egyáltalán szükség lenne. Egy jól megírt Web-alkalmazással lehet kezelni az adatbázist felhasználói oldalról.

felhasználó	felhasználó név	csoport tagja
adatbázis karbantartó 1 stb.	abkarb1	abkarb1, (mysql), users

8. táblázat - Felhasználók listája 3.

A következő fontos kérdés a **lemezkvóta**. Ha csak egy-két embernek van joga belépni a szerverre, akkor általában felesleges korlátozni az egy felhasználó által maximálisan

felhasználható helyet. Ha azonban több felhasználó is helyet kap a `/home` könyvtárban, akkor már hasznos lehet a korlátozás. Ha egy felhasználó esetleg teletömné a lemezt, pl. az `mp3` fájljaival, akkor a többiek számára nem maradna szabad hely a hasznos munkához. Az, hogy kit mennyire korlátozunk teljesen személyre szabható. Nem csak egyes felhasználókat, hanem csoportokat is szabályozhatunk. Minden olyan partíciót, ahova az átlagfelhasználónak írási joga van, érdemes kvótákkal ellátni. Esetünkben ez a `/home`, `/tmp` és a `/var/www` (a Web-csoport számára). Mivel az utóbbit valószínűleg csak munkára használja az ember, nem biztos, hogy korlátozni kell. A példában csak a `/home` partíciót korlátozom.

A kvóta két részből áll. Az első a *soft limit*, vagyis az a határ, melyet elméletileg szeretnénk, hogy betartsanak. A *soft limit*-et át lehet adott ideig lépni (*grace period* – türelmi idő), egészen a *hard limit* eléréséig. Ekkor új fájlokat a felhasználó már nem írhat a lemezre, az írni kívánt adatok elvesz(het)nek. Amint letelik a türelmi idő (általában 1 hét), a *soft limit* *hard limit*-té válik. Ilyenkor a felhasználónak le kell törölnie néhány fájlt, hogy visszatérhessen a felső határ alá.

Korlátozni a lefoglalt blokkokat és az *inode*-okat is lehet. A másodikra azért lehet szükség, mert egy rosszindulatú felhasználó sok rövid (pl. 0 bájtos) fájlt készíthet, mellyel lefoglalhatja az összes szabad *inode*-ot. Ekkor – bár a lemez nem telített – mégsem tudunk rá írni.

Az egyes felhasználók lemezkvóta-beállításait a *root* szabályozhatja az `edquota -u fnév` paranccsal. Egyes csoportok kvótái is szabályozhatóak: `edquota -g csnév`. Minden felhasználó ellenőrizheti a kvótáinak állását a `quota` paranccsal. A rendszergazda életét megkönnyíti a `repquota` parancs, mely egy adott partíció kvótáinak állásáról készít jelentést. A `quotatstats` program egy gyors statisztikát készít számunkra. A `quotacheck` leellenőrzi induláskor a kvótatáblázatokat. A `quotaon` és `quotaoff` parancsokkal lehet ki és bekapcsolni egy adott fájlrendszer kvóta-ellenőrzését. Bővebb információkért forduljunk a manuál oldalakhoz és a dokumentációhoz.

Példámban az alsó határt (*soft limit*) 20 megabájtban a felsőt (*hard limit*) pedig 40-ben jelölöm meg.

IV. Megvalósítás

A következőkben a Debian GNU/Linux 2.2 (Potato) változatának telepítését és behangolását fogom bemutatni lépésről lépésre. Ennek a fejezetnek az elolvasása inkább csak a rendszergazda-beállítottságú embereknek javasolt. Az egész fejezet elolvasásának csak akkor van értelme, ha egy számítógép mellett ülve végigkövetjük a teendőket. Azok számára, akik nem követik végig a feladatmegoldásomat, az egész fejezet értelmetlennek tűnhet, mert az anyag „másik fele” a számítógép monitorán jelenik meg (mivelhogy minden egyes képernyőképet nem tudok itt bemutatni).

Fontos feltétel, hogy addig, amíg az összes beállítást el nem végeztük a rendszeren és le nem teszteltük széleskörűen szerverünket, *NE* tegyük ki élesben az Internetre. A telepítést úgy végezzük, hogy a gép le legyen választva minden hálózatról. A tesztelést egy Internettől elzárt, belső hálózati szegmensben végezzük (mely nem része a produktív hálózatnak). Ekkor persze hogy lenne lehetséges az Internetről való telepítés? - kérdezheti az olvasó. Ha mindenképp ezt a megoldást választjuk, nem baj. Lényeg az, hogy a telepítés megkezdése előtt állítsuk be megfelelően a tűzfalat, nehogy menet közben rögtön ránk akadjanak. Továbbá a csomagok letöltése után kapcsolódjunk le a hálózatról. (Vagy töltsük le a csomagokat egy másik gépre, stb.)

1. Gyorstalpalás

A következőkben a „totálisan türelmetlenek” számára néhány lépésben felvázolom a Debian GNU/Linux Potato kiadásának telepítését. Később finomhangolom a rendszert.

1.1 A szoftver beszerezése: CD-set, vagy FTP tükör.

Mi sem egyszerűbb, mint letölteni a *CD ISO image* fájlokat (jelenleg 3 db kompakt lemez¹¹⁰) és kiírni őket. A lemezek elérhetőek többek között az ftp.fsn.hu alól is. Célszerű újraírható lemezre írni az anyagot. Ekkor, ha biztonsági frissítések látnak napvilágot, gyorsan átírhatjuk a lemezeket és nem kell kidobni azokat.

Ha nincs szélessávú elérésünk, kérjünk meg valakit a Linux-os levelező listákról, hogy írjon nekünk CD-t megegyezés szerint. Általában anyagáron meg szokták írni a lemezeket.

(Ne felejtsük el, hogy az első kompakt lemezről indíthatjuk a telepítést, ha az adott gép BIOS-a támogatja ezt. Ha nem, akkor legalább 2-5 floppy-t készítenünk kell.)

Ha viszont nem akarunk a CD-kel vesződni és van szélessávú Internet-elérésünk (értsd: legalább 1-2 Mbit/sec), akkor nyugodtan telepíthetjük a rendszert ftp-n keresztül

¹¹⁰ Vigyázat! A hivatalos Debian CD-k nem tartalmazzák a *non-free* szekciót. Ezért javaslom, hogy a Nagy Attila által készített „unofficial” CD image-eket töltsük le. Ezek a *non-US*-t is tartalmazzák.

is. Elég csak a *kernel* és a „*root*” floppy *image* fájlokat letölteni, és azokat floppyra másolni. A kernel floppyhoz tartoznak a *driver* lemezek. Ezeken a kernel moduljai találhatóak. Továbbá az alaprendszer is lemezeken van, méghozzá 11 db-on. Ha 1.44-es floppy meghajtót használunk, akkor összesen 16 db lemezre lesz szükségünk az alaprendszer telepítéséhez. Ha az alaprendszert FTP, NFS vagy HTTP protokollokon keresztül is be tudjuk szerezni, akkor elég 5 db floppy (*rescue*, *root*, *driver-1,2,3*).

A telepítéshez szükséges fájlok a `debian/dists/potato/main/disks-i386/current` könyvtárban találhatóak. (A telepítő lemezek magyar nyelvű fordítása megtalálható az [ftp://mlf.linux.rulez.org/pub/mirrors/debian-disks-hu](http://mlf.linux.rulez.org/pub/mirrors/debian-disks-hu) címen. A fordítás még nem teljes, fejlesztés alatt áll. A hivatalos tükrökön az angol nyelvű változat lelhető fel.) Először érdemes letölteni a `doc` alkönyvtárban lévő dokumentációkat és átolvasgatni azokat. A `doc/ch-hardware-req.en.html` fájl fontos információkat tartalmaz a szükséges és támogatott hardver eszközökről, továbbá a telepítéshez használt kernel és modulok milyenségéről. Ha ezt elolvassuk, sok fejfájást spórolhatunk meg.

Az installációs folyamathoz négyféle lemezkészletet készítettek. Minden készlet hozzáférhető egyben (*loadlin*-es változat), 1.2Mb, 1.44Mb és 2.8Mb-os floppy méretben is. A floppy *image* fájlok a megfelelő *image-méret* alkönyvtárakban lettek elhelyezve.

- A *standard*, általános célú készlet egyben megtalálható a fenti könyvtárban. Ez méretben a legnagyobb, szinte minden modul le lett fordítva. Ha nem tudjuk, hogy mire lesz szükség, ezt érdemes használni. Előnye, hogy szinte mindennel működik, amit a Linux támogat, hátránya a nagy mérete.
- A *compact* változat sokkal kisebb, csak egy modul lemez tartozik hozzá. Természetesen kevesebb hardvert támogat. Előnyös olyan esetekben, ahol tudjuk, hogy mire számíthatunk, és a megfelelő vezérlők benne vannak ebben a csomagban. Ez a készlet az azonos nevű alkönyvtárakban található. Olvassuk el a `README.txt` fájlt bővebb információkért.
- Az *idepci* készlet olyan esetben használatos, amikor nincsenek SCSI-s eszközeink (merevlemez) és PCI buszos IDE vezérlős merevlemezre akarjuk telepíteni a rendszert.
- Az *udma66* készletre akkor van szükségünk, ha a merevlemezünk ATA-66-os IDE vezérlőre van csatlakoztatva. (pl. HPT366)

Fontos kiemelni azt, hogy ha a gépen már van egy FAT típusú fájlrendszer, akkor elég letölteni a *linux* (kernel, 1Mb), *drivers.tgz* (modulok, 3.6Mb), *base2_2.tgz* (alaprendszer, 15Mb), *loadlin.exe* (kernel betöltő) fájlokat és a *root* floppy-t.

Ezután az `install.bat` indításával indulhat a kernel betöltése és a telepítés merevlemezről.

Ha floppy-s módszert válasszuk (mert pl. szűz merevlemezre akarunk telepíteni), de nincs még kéznél linux rendszer, akkor a `dosutils` könyvtárban lévő programokat letöltve (`loadlin`, `rawwrite`) segíthetjük az *image*-ek floppy-ra írását.

Linux alatt a `dd if=image of=/dev/fd0 bs=512; sync;` paranccsal írhatunk ki egy floppy-t. Ekkor persze a rajta lévő dolgok törlődnek. Fontos, hogy hibamentes lemezeket használjunk, mert nem lesz ellenőrizve a lemez írás közben, és esetleg a telepítés közepén derül ki a hiba.

Az `images-1.44` alkönyvtárban találhatjuk a `root.bin`, a `rescue.bin`, `drivers-x.bin` és a `base-x.bin` lemezeket 1.44Mb-os floppy meghajtóhoz. Ez a legelterjedtebb mostanában, a továbbiakban erre vonatkozik, amit írok.

1.2 A telepítés menete

Nézzük meg a telepítés konkrét lépéseit. Amikor már a kezünkben vannak a kész telepítő lemezek vagy CD-k, hangoljuk be a számítógép BIOS-át a nekünk legmegfelelőbb beállításra. Védjük le a *Setup*-ba való belépést jelszóval, a rendszerindítást viszont semmiképp se. Válasszuk ki indítási célként vagy a CD-ROM olvasót, vagy a lemezmeghajtót. A telepítés befejezése után ne felejtjük el ezt visszaállítani úgy, hogy az első indítható egység az a merevlemez legyen, amelyikre a rendszert telepítettük.

1.2.1 Indítás CD-ről vagy floppy-ról

Helyezzük be az indítható médiát és indítsuk el a számítógépet. Nemsokára ezt a képet láthatjuk: (a színeket a legtöbb képen megfordítottam az olvashatóság és a tinta kedvéért.)

Nyomogassuk végig az *F1..F10* billentyűket és olvassuk el az információkat. Ha valamely hardver eszköz vezérlőjének indulási paramétert kell átadnunk (pl. I/O bázis cím, megszakítás, stb.), akkor az tesszük meg. Általában a kernel mindent megtalál magától, és nem kell kézzel paraméterezni. Ha a hardver eszköz valamilyen nem hétköznapi I/O címen van, vagy a kernel modul / vezérlő nem ismeri fel magától, tájékozódjunk, hogy kell azt paraméterezni. Alább a SCSI kártyák minta paramétereit láthatjuk:

3. kép - Speciális indítási paraméterek

Az indításnak több lehetséges módoszata van:

4. kép - Indítási metódusok

Ha minden jól megy, mi csak nyomjunk le az *Enter* billentyűt – ekkor elindul a kernel betöltése és a hardver felismerése. A kernel betöltése után a rendszer jelez, hogy tesszük be a *root* floppy-t. (CD esetén erre nincs szükség). Ha minden jól ment, akkor egy üdvözlő képernyő fogad minket. Itt csak nyomjunk *Enter*-t.

1.2.2 Szükséges alapbeállítások, partícionálás

Először válasszunk billentyűzetet. Én az „*us*” billentyűkiosztást ajánlom. (És persze angol billentyűzetet, hiszen ezt a gépet nem szövegszerkesztésre fogjuk használni.)

Mivel még nincs Linux-os fájlrendszer a merevlemezén, partícionálnunk kell azt a tervünk alapján.¹¹¹

5. kép - Merevlemez partícionálás

Több lemez esetén ki kell választanunk azt, amelyiket fel akarjuk partícionálni. Jelen esetben ez a „*hda*” lesz. A következő képernyőkép figyelmeztet minket, hogy a LILO nem képes a régi merevlemezeken az 1023-as cylinder felett lévő részekről betölteni a kernelt. Nekünk ez itt nem számít, mert az első partíció a `/boot` lesz, így a kernel nem kerülhet azokra a területekre.¹¹²

Ha a lemez teljesen szűz, akkor egy kérdést kapunk, hogy új MBR táblát készíthet-e a rendszer. Természetesen válaszoljunk igennel. Ha a tábla esetleg hibás, akkor is ezt a képernyőt kaphatjuk. Esetünkben mindenképp töröljük az egész táblát, hiszen nem lesz más operációs rendszer a gépen. A `cfdisk` program segítségével feloszthatjuk a merevlemez. Ez egy elég könnyen használható és elég egyértelmű program (Bár én a mai napig jobban szeretem a fapados `fdisk` programot. Elszántaknak ezt az utóbbit ajánlom.)

6. kép - A cfdisk program

¹¹¹ Először is nem árt, ha elolvassuk ezt a részletes tanulmányt a partícionálás technikai oldaláról:

<http://www.vmg.sulinet.hu/vmg/home/szamtech/linux/node94.htm>

¹¹² A LILO legújabb változatában már kiküszöbölték ezt a problémát. Valószínűleg ez még nem lesz benne a Potato-ban.

(A fenti ábrán lévő méret-adatok irreálisak, mert ez egy virtuális gépen belüli telepítést mutat egy 280 MB-os lemezre.)

Hozzunk létre 3 db „*primary*” és 5 db „*logical*” szeletet (=partíciót) a „*New*” menüpontra lépve. A méreteket az eredeti terv arányai szerint válasszuk meg. A „*hda2*” szeletnél álljunk rá a „*Type*” menüpontra és válasszuk ki a 82-es kódot (*Linuxswap*). Amint minden kész, nyomjuk meg a „*Write*” menüpontot. Ekkor válaszoljunk „*yes*”-szel, igen tényleg ki akarjuk írni a táblát.

A következő lépés a swap partíció inicializálása. Válasszuk ki a megfelelő (esetünkben „*hda2*”) partíciót. A szeletről minden adat törlődik.

Most a szeletek formázása következik *ext2fs* fájlrendszerrel. Először a „*hda3*” részt válasszuk. Ne kérjünk 2.0-s kernel kompatibilitást, hiszen nem lesz rá szükségünk. A szeleten minden adat törlődik. A telepítő megkérdezi, hogy ez-e a gyökérnek szánt rész. Válaszoljunk igennel. Ezután sorra formázzuk meg a többi partíciót is hasonlóképpen. Amikor a partíció felfűzési pontjáról kérdez, jelöljük ki a helyes választ a tervünknek megfelelően. Ha a listában nem szerepel a pont, az „*other*” menüpontban megadhatjuk kézzel.

1.2.3 A hálózat beállítása

Ha ezzel készen vagyunk, telepíthetjük a kernelt és a modulokat a gépre. Válasszuk ki a floppy-t (vagy a CD-t) forrásmédiumként. Itt még nem lehet hálózatról telepíteni sajnos, mert még nincsen a hálózati kártya vezérlője betöltve, ami a floppy-n van.

Ha a floppy-t választottuk, tegyük be a „*rescue*” lemezt. A tartalma felmásolódik a gépre. Ha ez kész, a telepítő bekéri egymás után a három „*driver*” lemezt, mely a kernel moduljait tartalmazza.

7. kép - Modulok kiválasztása és betöltése a modconf programmal

Ami a hálózat működéséhez feltétlenül szükséges, azt keressük meg és töltsük be. (Pl. hálókártya vezérlő modulja) A „*cdrom*, *fs*, *ipv4*, *ipv6*, *video*” menüpontokban ne is keresgéljünk, szerverünkhöz szükséges vezérlők itt úgyse lesznek. Ha SCSI-s

merevlemezünk van, akkor azt már eddig úgyis fel kellett ismernie a kernelnek. A telepítéshez nem szükséges speciális eszközeinket pedig később is megkereshetjük a `modconf` program futtatásával.

Válasszuk ki tehát a „*net*” menüpontot. Itt keressük ki a hálókártyánknak megfelelő vezérlőt. Megkérdezi, hogy biztosan be akarjuk-e tölteni. „Igen”. Ezután – ha szükség van rá – paramétereket is adhatunk a moduloknak, mint pl. I/O bázis cím. Általában a legtöbb modul megtalálja az eszközt a szabványos címeken keresgélve. Ha a hálókártya ISA-PnP-s, akkor lehetőleg vegyük ki PnP-ből és „jumper”-oljuk fel egy adott megszakításra, különben a Linux nem ismeri fel egykönnyen.¹¹³

8. kép - Hálózati modulok tallózása

Ha a következő üzenetet kapjuk: „*Installation succeeded*”, akkor minden rendben, sikerült. Ha „*...failed*” a szöveg vége, akkor próbálkozzunk mással, vagy másik báziscímmel, megszakításokkal.

Ha már nincs más betölteni való eszközvezérlőnk, lépünk ki a `modconf`-ból és válasszuk a *Hálózat beállítása* menüpontot. Elsőként adjuk meg a szerver nevét. Ezt mindenki saját fantáziájára bízom. Legyen minél ötletesebb és ritkább (pl. egy szép női név). Én a példa számára az egyszerűség kedvéért az „*alfa*” nevet választottam.

Ha esetleg több hálókártya is lenne a gépben, akkor ebben a lépésben ki kell választani, hogy melyiket konfiguráljuk. Legyen az `eth0` eszköz.

Most az IP cím megállapításának módja következik. A telepítő felajánlja, hogy DHCP, vagy BOOTP protokoll segítségével szerez egy dinamikus IP címet. Nekünk ez nem jó, hiszen Web-szerverünknek statikus IP címe van. „Nem” a válasz. A következő kérdésre adjuk meg a statikus IP címünket, a hozzá tartozó hálózati maszk értékét, és az (Internet felé) átjáró IP címét. Ezek után a megvásárolt domén-nevet írjuk be. Esetünkben ez `boresszormegyar.hu`. A következő lépés a névkiszolgálók IP

¹¹³ A 2.4-es kernelben ezt már kiküszöbölték. (Lásd később)

címeinek megadása. Adjuk meg az elsődleges és másodlagos név-szerverek címait szóközzel elválasztva.

1.2.4 Alaprendszer telepítése, újraindítás a merevlemezről

Egy telepítési médiumot kell kiválasztani. Ha CD-lemezeink vannak, akkor semmi probléma, indulhat az alaprendszer telepítése. Nyomjunk párszor *Enter*-t. Az alapbeállítások célravezetőek.

Floppy esetén, ha az alaprendszert is hálózatról akarjuk letölteni (miért is ne?), akkor válasszuk ki a hálózatot. Ha valahol a Debian tükör ki van exportálva NFS segítségével, akkor azt is használhatjuk. Alapállásban egy HTTP címről szeretné letölteni az alaprendszert, mely messze Amerikában található. Szerencsére a <http://ftp.fsn.hu/ftp:80> alatt találunk megfelelő magyar tükört. Írjuk be ezt a cím helyett. Ha az alaprendszert is kiírtuk floppy-ra, és nem akarunk / tudunk hálózati telepítést, akkor egyenként tegyük be a lemezeket.

Miután az alaprendszert sikerült telepíteni, a rendszer beállítása következik. Itt meg kell adni az időzónánkat. Válasszuk ki a *CET*-et (*Central European Time*). Ekkor kérdést kapunk a hardver óra felől. Mivel csak ez a rendszer lesz a gépen, állítsuk a gépidőt a *GMT*-hez. (Ez a 0-s időzóna).

A gép merevlemezének indíthatóvá tétele a következő feladat. A LILO-t tegyük az MBR¹¹⁴-be, vagyis válasszuk az első lehetőséget. Ha van még egy üres floppy-nk, akkor készíthetünk egy *boot-floppy*-t. Mivel a *rescue* floppy segítségével is be tudunk jutni a rendszerbe, ezt elhagyhatjuk.

Vegyük ki a telepítő médiumot a meghajtóból, nehogy az induljon el a merevlemez helyett! A felmerülő „tényleg mehet-e az újraindítás” kérdésre ezután válaszoljunk igennel.

1.2.5 A jelszórendszer beállítása. („MD5”, „Shadow password”)

Miután a rendszerünk felállt, megkérdezi a telepítő, hogy akarunk-e MD5-ös jelszókódolást. Természetesen akarunk, hiszen ekkor maximum 8 karakter helyett maximum 127 karakteres jelszókat is használhatunk, ami nagyban növeli a biztonságot. Ezután válaszoljunk szintén igennel, akarunk árnyék-jelszófájlt (*shadow*), ez is közelebb visz a biztonsághoz.

A következő lépés a rendszergazdai (*root*) jelszó megadása. Mostanra már ezt is kitaláltuk a tervünk szerint.¹¹⁵ Írjuk be kétszer.

¹¹⁴ Master Boot Record, a merevlemez első 512 bájtja.

¹¹⁵ Legjobb, ha a *spwgen -s*, vagy az *spwgen* parancsokkal készítjük el a jelszavakat.

Hozzunk létre legalább egy felhasználót, még hozzá a rendszergazdát, az „*rgazda*” nevűt. (A telepítő segítségével.) Adjuk meg a nevet, a valódi nevet és kétszer a jelszót. Most a telepítő észrevette, hogy a PCMCIA modulokat nem is használjuk, nincs ilyen eszköz a gépben. Bátran távolíttassuk el vele.

1.2.6 Az „apt” program beállítása

A következő lépés az apt program letöltési forrásának beállítása. Itt kell megadnunk, hogy a Debian tükör (a CD is annak számít valamilyen mértékben) hol található. Innen fogja leszedni a csomaglistát. Ezután kezdetünk csak neki a csomagok kiválasztásának. CD-s telepítés esetén válasszuk ki a „*cdrom*” menüpontot és az alapértelmezett beállítások kiválasztásával biztosan sikerrel járunk. Floppy-s telepítés esetén válasszuk az „*ftp*” menüpontot.

9. kép - Az APT program beállítása

Ekkor kérdést kapunk: akarunk-e a non-US tükrökről származó (vagyis kriptográfiát tartalmazó) programokat használni. Mivel a szervergép nem az USA-ban helyezkedik el, válaszoljunk „igen”-nel. A következő kérdés a non-free, majd a contrib szekciókra vonatkozik.¹¹⁶ Itt is válaszoljunk igennel.

Ekkor a Debian tükrök listáját kínálja fel. Először válasszuk ki Magyarországot (Hungary), aztán a hozzánk sávszélességben közelebb eső szerveret.¹¹⁷ Én az ftp.hu.debian.org-ot választom.

Biztos, ami biztos, a legjobb, ha az apt-t saját kézzel konfiguráljuk, ugyanis ekkor megadhatjuk a biztonsági frissítéseket tartalmazó könyvtárat is. Válasszuk ki az „*Edit sources by hand*” menüpontot. Ekkor egy egyszerű és jól használható szövegszerkesztő jön elő (ae). Módosítsuk a rendszert a következőképp:

```
deb ftp://ftp.hu.debian.org/debian potato main contrib non-free
deb ftp://ftp.hu.debian.org/debian-non-US non-US/main non-US/contrib non-US/non-free
deb ftp://ftp.hu.debian.org/debian dists/potato-proposed-updates/
```

¹¹⁶ Részletesen a II. Alapfogalmak / 3. A Debian projekt c. fejezetben.

¹¹⁷ Ha hozzánk sávszélességben közelebb esik egy itt fel nem sorolt, de megbízható Debian tükör, akkor kézzel kell megszerkesztenünk a beállításokat.

Az első sorban a „fő” debian szerveret jelöljük meg, melyen nincsenek kriptográfiát tartalmazó programok. Itt három részre oszlik a rendszer: a fő, a nem-szabad és a nem szabadhoz kapcsolódó programokra. A második sorban ugyanez igaz, de a titkosítást tartalmazó programokra. A harmadik sor a biztonsági frissítések külön könyvtára.

A sorok jelentése: *csomag protokoll://szervernév/tükörgyökér változat szekciók*

- *csomag*: *deb*: bináris csomag, vagyis a *.deb* fájlok kellene (forráskód esetén „*deb-src*”)
- *protokoll*: lehet „*ftp*”, „*file*”¹¹⁸ vagy „*http*”
- *szervernév*: a tükör helye, domén-név, vagy IP cím
- *tükörgyökér*: az adott szerveren belül hol kezdődik a tükör (melyik alkönyvtárban)
- *változat*: lehet „*stable*” „*unstable*” „*frozen*”. Ez mind a három egy-egy szimbolikus kötés (link) az adott állapotban lévő változathoz. Az írás pillanatában a Potato még „*frozen*” állapotban van¹¹⁹, ezért inkább direkt módon határozom meg annak a helyét.
- *szekciók*: a használni kívánt szekciók egymástól üres közzel elválasztva.

Mentsük el a változtatásokat (a képernyő tetején segítség olvasható, a „^” jel a *Control* billentyűt jelenti.) Ekkor az *apt-get* program letölti a csomaglistát a szerverről. Ha ez sikerült mehetünk tovább. Ha nem, akkor szerkesszük át a forráslistát.

Most azt kell eldönteni, hogy a csomagokat egyenként (*advanced*) válogatjuk ki, vagy egy előre elkészített összeállítást használjunk. Én az egyenkénti kiválasztást javaslom. Igaz ez sokáig eltarthat, de így pontosan meg tudjuk határozni, hogy mi kell és mi nem. A türelmetlenek válasszák ki a *Web-server* pontot. Ekkor egy általános csomaglista alapján kerülnek telepítésre, ami eléggé különbözik az általam felsoroltaktól.

Később

- az *apt-get update* paranccsal frissíthetjük a csomaglistát,
- az *apt-get upgrade* paranccsal frissíthetjük a csomagokat,
- az *apt-get dselect-upgrade* paranccsal a *dselect* által kiválasztott új csomagokat is telepíthetjük,
- az *apt-get install csomagnév* paranccsal letölthetünk és telepíthetünk egy csomagot
- az *apt-get remove csomagnév* paranccsal eltávolíthatunk csomagokat
- és még sok mást is tehetünk, ha elolvastuk a dokumentációt.

¹¹⁸ Pl. NFS vagy CD esetén.

¹¹⁹ Bár eredetileg 2000. márciusára tervezték, 2000. augusztus 15.-én jelent meg a Potato „stable” változata.

1.2.7 A „dselect” program

A `dselect` program használata igen egyszerű és egyértelmű, ha az ember már ismeri. Más rendszerekhez szokott embernek ez először nagyon ijesztő lehet, mert ez egy igazi UN*X-os szemléletű program. Először is olvassuk el a súgóját, hogy mit hogyan kell csinálni, mert különben nem megyünk semmire. (A súgó a 2. menüpont alatt érhető el.)

10. kép - dselect - Főmenü

A főmenü 7 pontból áll.

- *0. Access:* ki tudjuk választani, hogy milyen médiáról telepítjük a csomagokat. Ez a módszer lehet most *apt*, *floppy* és *nfs*. (CD esetben *cdrom*, *multicd* is), továbbá később lehet *http*, *ftp* is. Mivel az *apt* már be van állítva, ezzel ne is foglalkozzunk.
- *1. Update:* itt indíthatjuk a csomaglista frissítését. Esetünkben ez az *apt-get update* parancsnak felel meg.
- *2. Select:* Ez az egész program szíve-lelke. A csomaglistában tallózhatunk, és a kívánt csomagokat kijelölhetjük telepítésre, megtartásra, törlésre, vagy teljes törlésre.¹²⁰ Részletesen később.
- *3. Install:* Ezzel indíthatjuk a csomagok letöltését (Internet esetén) vagy bemásolását (CD esetén). Miután a csomag a rendszerbe került, ki lesz csomagolva, majd be lesz állítva. Bizonyos csomagok interaktivitást igényelnek a rendszergazdától beállítás közben.
- *4. Config:* Ha egy csomagot az előző menetben nem tudott a rendszer beállítani, de már ki lett bontva, akkor itt újra próbálkozhatunk.
- *5. Remove:* A törlésre jelölt csomagok eltávolítását itt lehet elindítani.
- *6. Quit:* A `dselect` programból való kilépés.

¹²⁰ Az II. Alapfogalmak / 3. A Debian projekt c. fejezetben bővebben van ismertetve ez a téma.

Belépve a „*Select*” menübe egy üdvözlő képernyőt kapunk, mely tájékoztat a program használatáról. Ezt itt olvassuk végig. Nyomjuk meg a „?” gombot, majd a „k” betűt. Ekkor az összes felhasználható funkció és a hozzá tartozó billentyű fel lesz sorolva.

11. kép - dselect - Súgó

Navigálni a csomagok között a kurzorgombokkal lehet. Ha egy csomagot telepíteni akarunk, jelöljük meg a „+” gombbal. Ha törölni akarjuk, akkor jelöljük meg a „-” gombbal. Ha azt akarjuk, hogy a csomaghoz tartozó konfigurációs fájlok is törölődjenek, akkor jelöljük meg a „_” gombbal. Ha azt akarjuk, hogy a csomag ne frissüljön, akkor jelöljük meg a „=” gombbal. Ha később mégis azt szeretnénk, hogy frissüljön a csomag, jelöljük meg a „:” gombbal.

A csomagokat többféle szempont szerint is sorba lehet rendezni. Nyomjuk meg az „o” gombot egyszer, az „O”-t pedig kétszer. Ekkor szekció szerint fogjuk rendezni a csomagokat. Véleményem szerint telepítéskor ez a legjobb sorrend, karbantartáskor viszont csak egyszer nyomjuk meg az „O”-t (ekkor aszerint is rendezi, hogy az adott csomag telepítve van-e már vagy nincs).

A csomaglistából az Enter leütésével lehet kilépni, ezt eleinte nehéz megszokni. „Q”-val felülbírálnak a csomagfüggőségeket. Esc-el pedig a változtatások elhagyásával léphetünk ki.

A csomagnevek között keresni a „/” gomb lenyomásával lehet. Ha ezt a nevet akarjuk továbbra is kerestetni, akkor a „\” gombbal megtehetjük.

A súgóból a Space billentyűvel léphetünk ki. Először lehet, hogy nem érthető mi az a három „*” vagy három „-”, egymás mellett. Nyomjuk meg a „v” betűt és rögtön megértjük.

A csomagok kiválasztása

Miután beléptünk a csomaglistába először is nyomjuk meg az „o”-t majd kétszer az „O”-t. Ezután a „/” gombot leütve keressük meg egyenként a következő csomagokat és jelöljük meg őket telepítésre a „+” gombbal. Ha egy csomag egy másiktól függ, egy új képernyő fog megjelenni. Ettől ne ijedjünk meg. Ha valami függ egy másiktól

(*depends*), akkor a program automatikusan megjelöli és megkér minket, hogy fogadjuk el ezt a beállítást. Ha mégse tetszene a függés okozta változás, akkor nyomjuk meg az „R” gombot. Ha megfelel így is, akkor nyomjunk Enter-t. Ha csak javasolja a csomag egy másik csomag telepítését is (*recommends*, *suggests*), akkor ő nem jelöli meg, a döntést teljesen ránk bízta. Javaslatom, hogy tartsuk magunkat a következő listákhoz.

```
„+”: cruft, debconf, logcheck, vlock, members, memstat, quota, slay, suidmanager,
syslog-ng, syslog-summary, systune, tmpreaper, whowatch, hdparm, watchdog, slocate,
kernel-image-2.2.16, dpkg-mountable, dpkg-multicd, doc-base, manpages-hu, joe,
postfix-tls, libssl09, openssl, screen, ud, ippl, snort, traceroute, tripwire, mysql-
client, mysql-server, apache-ssl, apache-common, ssh, cracklib-runtime, cracklib2,
wipe, analog, php3, php3-mysql, webalizer, wget
```

12. kép - dselect - Csomaglista

A következő csomagok rövid összefoglalóját olvassuk el, és igény / hardver konfiguráció szerint válasszuk ki azokat telepítésre, melyek számunkra szükségesek, vagy hasznosak. Semmiképp se válasszuk ki mindet, mert több olyan csomag is van, melyek hasonló, vagy ugyanazon funkciót töltik be¹²¹, továbbá egyes csomagok csak speciális hardverelemek esetében szükségesek.

```
esetleg „+”: anacron, linuxconf, mon, makepasswd, pwgen, raidtools, raidtools2, sudo,
alien, apcd, autolog, bpowerd, ext2resize, mtx, genpower, powstatd, timeoutd, upsd,
apache-doc, debian-guide, dhelp, doc-rfc, dpkg-www, dwww, info2www, sysadmin-guide,
ftape-doc, grep-mail, pkg-order, eject, ncftp, netcat, linuxlogo, lm-sensors, bing,
echoping, fping, fwctl, icmpinfo, netmask, netselect, ntop, queso, lshell, rdate,
tcpdump, nstreams, asp, mason, netdig, nmap, , libapache-mod-auth-pam, wdsetup, idled,
doc-html-w3, nwm, cronolog, ldp-nag, mysql-doc, tdlug, wdg-html-ref, lasg, bigbrother,
gnupg, lynx-ssl, zip-crypt, unzip-crypt, powstatd-crypt, cdrecord, mkisofs, afbackup,
afbackup-client, amanda-*, apcupsd, bzip2, dlocate, dump, hwttools, kbackup-*, knl,
ltrance, mc, mc-common, parted, setcd, sformat, symlinks, sysutils, taper, tob, tree,
vfu, yard, fonty, ftape-util, set6x86, statserial, weblint, zope-*, wml, linbot, www-
mysql
```

¹²¹ Ne feledjük, hogy a dokumentáció nagy része HTML formátumú. Ha a web-szerveren akarjuk ezeket olvasgatni, tegyük fel egy HTML böngészőt. Javasolom a lynx-ssl csomagot. A teljes dokumentációt ki is szolgáltatathatjuk a Web-szerveren keresztül. Ezt csak teljes körültekintéssel tegyük! Kérjük felhasználó-azonosítást ehhez is, hiszen így információk szerezhetőek arról, milyen programok futnak a gépen.

A következő csomagokat pedig jelöljük ki eltávolításra („_”), vagyis - most még nem-telepítésre. A zárójelben lévő csomagokat én törlésre ajánlom, viszont mások esetleg hasznosnak találhatják, ezért tessék elolvasni a hozzájuk tartozó információt és igény szerint választani.

```
„_”: bison, flex, dpkg-perl, tetex-bin, bin86, gcc, fbset, (elvis-tiny), ppp,
pppconfig, g++, libstdc++-210-dev, (isapnptools), pump, (fdutils), xviddetect, tetex-
base, gdb, libc6-dev, make, dpkg-dev, rcs, manpages-dev, (nvi), emacs20, emacs-common,
(perl-5.005-doc), (perl-5.005-suid), m4, libindent, exim, libopenldap1, libopenldap-
runtime, libpcre2, libblockfile1, libpng2, tetex-lib, dialog, (mutt), (procmail), dc,
bc, gpm, fingerd, lpr, nfs-common, nfs-kernel-server, pidentd, talk, talkd, telnetd
```

Ha ezekkel végeztünk, nyomjunk Enter-t és ekkor kikerülünk a főmenübe.

Indítsuk el az „*Install*” menüpontot. Ez a parancs megfelel az `apt-get dselect-upgrade` parancsnak. Az én konfigurációm esetében mindössze 35 MB-nyi csomagot kell letölteni. Kibontás után kb. 55 MB helyet foglal a rendszeren.

1.2.8 A feltelepített programok konfigurálása

Miután az összes kért csomag lejött a tükörről, a rendszer megkérdi, hogy a programokat milyen felületen szeretnénk beállítani. A választási lehetőségek a következők: *Dialog* (dialógus ablakok), *Text* (hagyományos, egyszerű szöveges felület), *Web* (böngészővel), *Noninteractive* (mindenből az alapbeállítást tárolja el, később kézzel beállíthatjuk, amit akarunk. Bár én legjobban a sima szöveges módot szeretem, a kezdők kedvéért a barátságosabb dialógusos módszert tárgyalom a következőkben.

`debconf` beállítása: A következő kérdés az, hogy milyen szintű kérdések alatti prioritású kérdésekkel nem akarunk foglalkozni: *medium*, *critical*, *high*, *low*. Válaszunk legyen „*low*”, tehát minden kérdést megválaszolunk. A következő kérdés arra vonatkozik, hogy az adott csomag kérdéseire adott válaszunkat megjegyezze-e és azt válaszolja automatikusan minden újabb frissítésnél. Most még válaszoljunk nemmel, hiszen lehet, hogy valamit elrontunk. A kérdés furfangos: megmutassam-e újra és újra a régi kérdéseket: Igen. Később ezt javasolt Nem-re változtatni a `dpkg-reconfigure debconf` parancs segítségével.¹²²

A csomagok felépítésükben hasonlítanak a `.tar.gz` fájlokra, telepítő / eltávolító script-ekkel vannak ellátva, stb. Egy csomag telepítése két részből áll: először kibontja a csomagból a fájlokat, és a megfelelő helyre másolja őket (*unpacking*). A második szakaszban pedig beállítja a konfigurációs állományokat és futtatási jogot ad a

¹²² Ez a kezdők kedvéért „igen”. Akik már jól ismerik a rendszert mindenképp válaszoljanak „Nem”-el, mert nagyon idegesítő, ha mindent megismétel. Később minden `debconf` dialógussal ellátott csomagot újra be lehet állítani a `dpkg-reconfigure csomagnév` paranccsal.

futtatható fájloknak (*setting up*). Ha a beállításhoz szükség van a rendszergazda döntésére is, akkor megkérdezi.

A következőkben az én általam összeállított csomaglista beállító kérdéseit sorolom fel. Más lista (csomagok) esetén más kérdések lehetnek.

Az „*unpacking*” fázisban:

netbase:

1. adjuk meg, hogy mely IP tartomány helyi: 127.0.0.1/8 (mivel igazi IP címünk van, nem adunk meg belső hálózati tartományokat.)

2. adjuk meg mely hálózati interfészekkel rendelkezünk, pl.: eth0 eth1

logcheck: Engedélyezzük, hogy felülírja az /etc/cron.d/logcheck fájlt, ha ezt kéri.

mysql-server:

1. figyelmeztet, hogy adjunk meg egy adatbázis rendszergazda felhasználót. Ezt majd később megtesszük.

2. Megkérdezi, hogy ha a mysql-server teljes eltávolítását kérjük, akkor letörölje-e az adatbázisainkat is. Válaszoljunk nemmel.

snort: (portscan-detektor): adjuk meg azt az IP tartományt, ahonnan portscan támadásokat várunk. Mivel az interneten vagyunk, ez legyen a saját IP címünk/tartományunk. (Vagy az Internet Kijárat IP-je, stb.)

ssh: az ssh kliens kapjon-e SUID bitet. Semmiképp, mivel nem akarunk .rhosts azonosítást. Nem a válasz.

lilo: megkérdi, hogy az új LILO fájlok segítségével hozzon-e létre új indítót. Igen.

A következőkben számos csomag kerül kibontásra és telepítésre. Ezek nem igényelnek interaktivitást.

A következő kérdés az /etc/motd (Message of the day) cseréje. Y,I: igen, N,O: nem, D: megmutatja a kettő közötti különbséget, Z: majd később döntök. Válasszuk az Y-t. A csomagok telepítése folytatódik.

A „*setting up*” fázisban:

netbase:

1. kikapcsoljuk-e az inetd.conf-ban a DoS veszélyes szolgáltatásokat? Igen.

2. Az ipfwadm parancs legyen-e ipchains kompatibilis átjáró? Igen.

3. Akarunk-e IPv6-os címeket az /etc/hosts-ban. Nem.

apache-ssl: Az SSL szerverhez készül egy azonosító. A cégünkről kell adatokat megadnunk, hogy majd a kliens azonosíthassa szerverünket. Írjuk be az ország nevét (HU), a megye nevét (pl. Zala), a város nevét, a cég nevét, az eszköz titulusát: Web-szerver, majd végül a szerver nevét: www.boresszormegyar.hu, email címét (pl. info@boresszormegyar.hu).

postfix-tls:

1. kérdést tesz fel, hogy a levéltovábbítás előtt az átmeneti fájlok olvashatóak legyenek-e a gépen fenn levő felhasználók által (gyorsabb út), vagy legyen egy külön felhasználó (postdrop) létrehozva, és ekkor csak a rendszergazda láthatja ezeket a fájlokat (lassabb, de biztonságosabb út). Válaszunk Igen.

2. Adjuk meg a gépünk nevét: *alfa.boresszormegyar.hu*

3. Elindítsa-e a levelező démont? Igen.

13. kép - a "snort" program beállítása

snort:

1. Mikor induljon el a portscan detektor? „boot” vagyis induláskor.
2. Melyik interfészen figyeljen? Amelyiken az Internetre vagyunk kapcsolódva. Nekem eth0.
3. A hálókártya hallgatózó üzemmódját kikapcsolja-e? Igen.¹²³
4. A következő kérdésre Igen legyen a válasz, itt nem részletezem.
5. Addicionális paraméterek: nyomjunk egy Enter-t.
6. Ki kapja meg email-ben a napi statisztikákat: adjuk meg e-mail címünket.
7. Itt is nyomjunk Enter-t

webalizer:

1. hova tegye a kész statisztikákat? Ha ki akarjuk tenni a Web-re, akkor nyomjunk Enter-t (nem ajánlott). Adjunk meg neki egy nem nyilvános könyvtárat.
2. Melyik gépnek nézzük meg a statisztikáit? Adjuk meg az egyik virtuális Web-szerverünk nevét.

php3: elindítsam az apache-ssl beállítóját, hogy a modult betöltse? Igen.

apache-ssl: ki a Web-szerver rendszergazdája (e-mail). Az alapértelmezés jó lesz, lásd későbbiekben.

3. Mi legyen a nyilvános neve a Web-szervernek? www.boresszormegyar.hu
4. megkeresi a dinamikusan betölthető modulokat. Ezeket majd később beállítjuk kézzel. Elementsem a beállításokat? Igen.
5. Újraindítam az Apache-ot? Igen.

lshell: (korlátozott *shell*, az átlagfelhasználók jogait tudjuk vele korlátozni belső DoS támadások kivédésre) A kérdésre válaszoljunk Igennek. Figyelem! Azoknak a felhasználóknak, melyeknek sok erőforrást biztosítunk, adjuk vissza később a `/bin/bash` rendes *shell*-t az `/etc/passwd` fájlban!

lynx-ssl: adjuk meg kezdőoldalnak a saját gépünket.

tripwire: (fájl integritás auditáló program) kinek küldje el a rendszeren észlelt változásokat? Adjuk meg a gyakran olvasott e-mail címünket.

watchdog: (hasznos őrszem DoS ellen): elindítsa-e minden induláskor a démonját? Igen. És most? Igen.

¹²³ Bár ekkor a többi gép felé irányuló portscan-eket nem figyeli meg, de nem is ez a feladata. Ha bekapcsolnánk a hallgatózó üzemmódot, az nagyban lassítaná az Ethernet kártyát.

2. Finomítás

Természetesen az eddig elvégzettek finomhangolásra szorulnak. Egy rendszer akkor van készen, ha optimalizáltuk az adott hardverhez és az adott igényeinkhez. A finomítás elég időigényes is lehet, de hosszú távon megéri. A finomhangolás legfőbb célja a sebesség és a biztonság fokozása. Persze a kényelem se az utolsó szempont. (Ez alatt nem a biztonság hiányát, vagy háttérbeszorítását értem. Hiszen maga a biztonság is kényelmet nyújt.)

Kezdetnek – a következő lépések magyarázatára - elolvashatjuk ezt a cikket [21].

2.1 Első lépések

1. Első lépésként tegyük a `/root` könyvtárat **olvashatatlan**á mások számára: `chmod o-rx /root` Tegyük meg ezt továbbá minden felhasználói könyvtárral is, hogy a felhasználók egymás személyes anyagába ne nézhessenek bele: `chmod o-rx /home/*`¹²⁴
2. Állítsuk be a **TCP SYN-flood** elleni védekezést¹²⁵: szerkesszük meg az `/etc/network/options` fájlt és írjuk be ezt a sort: `syncookies=yes`, továbbá egy ilyen sornak is szerepelnie kell: `spoofprotect=yes`¹²⁶ Ezután állítsuk le, majd indítsuk újra a hálózati interfészt: `/etc/init.d/networking stop; /etc/init.d/networking start`
3. Változtassuk meg az `/etc/default/rcS` fájl utolsó sorát **FSCCKFIX=no** -ról **yes**-re. Ezzel elérjük, hogy ha a rendszer hibásan állna le és az `fsck` program beindul, annak minden kérdésére válaszoljon **yes**-el a rendszer, különben rendszergazdai beavatkozásra lenne szükség a helyszínen. (Vagyis minden felmerülő hibát automatikusan kijavít.)
4. Készítsük el a következő **ipchains** táblát, pl. a `/root/szabalyok.sh` fájlba:

```
#!/bin/sh
MYIP="sajátIP"
# IPChains szabályok
# Az alapértelmezett viselkedés tiltó (-P = Policy)
ipchains -P input DENY
ipchains -P output DENY
ipchains -P forward DENY
# Kitöröljük az előző szabályokat (-F = Flush)
ipchains -F
# Befelé jövő kérések (-A = Add, -p protocol, -i interface, -s source, -d destination)
# Ezekon a portokon lehet kérni szolgáltatást
ipchains -A input -p tcp -i eth0 -j ACCEPT -s 0.0.0.0/0 -d $MYIP 80
# a többihez nem írom oda a -s 0.0.0.0/0 (bárhonnan) -t mert ez az alapértelmezés
ipchains -A input -p tcp -i eth0 -j ACCEPT -d $MYIP 443
```

¹²⁴ Ezt a lépést megtehetjük az összes felhasználó létrehozása után is.

¹²⁵ Ezzel egy DoS típusú támadás ellen védekezhetünk. A támadó SYN jelekkel árasztja el a portjainkat, de a forrás IP-je hamis.

¹²⁶ Ekkor az `/etc/init.d/networking` script IP átejtés ellen védekező `ipchains` szabályokat fog létrehozni.

```
ipchains -A input -p tcp -i eth0 -j ACCEPT -d $MYIP 22
ipchains -A input -p tcp -i eth0 -j ACCEPT -d $MYIP 25
# Ahhoz, hogy a belső gépről is tudjunk kapcsolódni kifelé, szükség van egy
# befelé jövő kapcsolati portra is. Mivel ez tetszőleges lehet, ezért tiltsunk
# le minden olyan csomagot, mely kapcsolódni próbálna a portjainkra (-y = SYN flag)
ipchains -A input -p tcp -y -i eth0 -j REJECT -d $MYIP
# itt viszont megnyitunk minden portot (ACK, FIN, ACK+SYN flag-es csomagok jöhetnek)
ipchains -A input -p tcp -i eth0 -j ACCEPT -s 0.0.0.0/0 -d $MYIP

#Megengedünk néhány ICMP típust a helyes működés érdekében
# A "nagyok" javaslata alapján ezeket szabad beengedni:
# echo-reply (echo-request) time-exceeded destination-unreachable source-quench
# (a következőket egyenként egy sorba kell írni)
ipchains -A input -p icmp -i eth0 -j ACCEPT -d $MYIP --icmp-type echo-reply
ipchains -A input -p icmp -i eth0 -j ACCEPT -d $MYIP --icmp-type echo-request
ipchains -A input -p icmp -i eth0 -j ACCEPT -d $MYIP --icmp-type time-exceeded
ipchains -A input -p icmp -i eth0 -j ACCEPT -d $MYIP --icmp-type destination-
unreachable
ipchains -A input -p icmp -i eth0 -j ACCEPT -d $MYIP --icmp-type source-quench

# a localhost-nak megengedhetjük a működést az lo interfészen
ipchains -A input -p all -i lo -j ACCEPT -s 127.0.0.0/8 -d 127.0.0.0/8
ipchains -A output -p all -i lo -j ACCEPT -s 127.0.0.0/8 -d 127.0.0.0/8
# viszont meg kell tiltanunk azt, hogy valaki a 127.0.0.0-s tartományból
# küldjön csomagot a nem-hurok interfészekre:
ipchains -A input -j DENY -l -s 127.0.0.0/8 -i ! lo

# a kifelé mehet minden, tcp, udp, icmp, stb.
ipchains -A output -p all -i eth0 -j ACCEPT -s $MYIP -d 0.0.0.0/0
```

A fenti lista feltételezi, hogy csak egy Ethernet interfészünk és egy IP címünk van. Több hálózati kártya és IP cím esetén értelemszerűen módosítsuk, vagy egészítsük ki a listát. Ha nem gépeltünk el semmit, és a teszt is működőképesnek mutatja a rendszert, akkor szerkesszük meg. Ha kész, futtassuk le a fájlt: `sh /root/szabalyok.sh` majd tároltassuk el a beállításokat az `ipchains-save > /etc/default/ipchains.rules` paranccsal. A szerver leállása esetén ezt a listát induláskor újra be kell tölteni. Lényeges, hogy a lista még a hálózati interfészek felhúzása előtt töltődjön be. Ezt megtehetjük úgy is, hogy átszerkesztjük a `networking` script-et, vagy írunk egy egyszerű indító script-et `/etc/init.d/ipchains-rules.sh` néven.

```
#!/bin/sh
echo "Restoring IPChains rules..."
/sbin/ipchains-restore < /etc/default/ipchains.rules
```

Tegyük a fájlt futtathatóvá: `chmod a+x /etc/init.d/ipchains-rules.sh` Hogy induláskor ez el is induljon, egy szimbolikus linket kell elhelyeznünk az `/etc/rcS.d` könyvtárban:

`ln -s /etc/init.d/ipchains-rules.sh /etc/rcS.d/S38ipchains-rules.sh` (Ugyanis a networking a 40-es számot viseli.)

5. Szerkesszük meg az `/etc/host.conf` fájlt a következőképp:

```
order hosts,bind      # mi a név-lekérdezés sorrendje
multi on              # egy névhez az összes hozzá tartozó IP-t adja vissza
nospoof on            # a DNS átejtést próbálja kiküszöbölni127
spoofalert on         # az átejtési kísérletet naplózza
```

6. `ippl`: *IP Protocol Logger*, vagyis egy csomagforgalom naplózó. Később segítségünkre lehet betörés utáni nyomozásra. Olvassuk el a manuált¹²⁸, szerkesszük meg az `/etc/ippl.conf`-ot, majd indítsuk újra a démont (`/etc/init.d/ippl restart`).

```
runas nobody          # Milyen jogokkal fusson
noresolve all         # nem kell DNS feloldás, mert lassít
# ident               # nem kell ident kikeresés, mer ez is lassít
# Alapesetben a syslog()-on keresztül naplóz, de
# a három protokollt külön logfájlba is helyezhetjük
#log-in tcp /var/log/ippl/tcp.log
#log-in udp /var/log/ippl/udp.log
#log-in icmp /var/log/ippl/icmp.log
run icmp tcp          # csak az icmp-t és a tcp-t naplózzuk
logformat normal all  # a log bőeszédősége: short / normal / detailed
ignore icmp type echo_reply # a ping csomagokat nem naplózzuk
log options ident,resolve tcp port 22 # az ssh-s kapcsolatokat lenyomozzuk
```

Ahhoz, hogy szét tudjuk válogatni az `ippl` üzeneteit, helyezzük el ezt a három sort az `/etc/syslog-ng/syslog-ng.conf`-ba, majd indítsuk újra a `syslog-ng`-t.

```
destination ippl { file("/var/log/ippl.log"); }; # ide fognak kerülni az üzenetek
filter f_ippl { program(ippl); }; # ennek a szűrőnek a segítségével
log { source(src); filter(f_ippl); destination(ippl); };
```

7. Ha szükség van rá, állítsunk be a kernelnek számunkra megfelelő értékeket a `systemd` program segítségével (ha telepítve lett.) Szerkesszük meg az `/etc/systemd.conf` fájlt.

```
# 2.2 és 2.4 kernelek beállításai
# az alapértelmezett egyszerre nyitott fájlok száma 4096
# Amennyiben TÉNYLEG szükség van rá a nagy forgalom és a sok fájl
# miatt, (olvassuk el először a dokumentációt!) az értéket megnövelhetjük így:
# /proc/sys/fs/file-max:16384
# a következő paraméter hasonlóképpen szabályozza az inode számokat:
# /proc/sys/fs/inode-max:16384 # eredetileg 8192
```

Ha változtattunk valamit a beállításokon, érvényesítsük ezeket:

`/etc/init.d/systemd restart`

¹²⁷ Bővebben: `man host.conf`

¹²⁸ `man ippl.conf`, `man ippl`, `/usr/doc/ippl/*`

8. Mentsük el floppy lemezre a partíciós táblát. Ez nagyon előnyös lehet később, ha esetleg valamilyen hiba miatt megsérülne az.

```
mount /floppy
```

```
dd if=/dev/hda of=/floppy/mbr.gépnév.dátum bs=512 count=1
```

Visszatölthetjük később ezzel a paranccsal:

```
dd if=/floppy/mbr.gépnév.dátum of=/dev/hda bs=1 count=64 skip=446 seek=446
```

9. Készítsünk egy **mysql** rendszergazda¹²⁹ jelszót, mely különbözzön a rendszergazdai jelszótól: `mysqladmin -u root password új-jelszó`
Ezután végezzük el a következőket:¹³⁰

- hozzunk létre szövegszerkesztővel egy `/root/.my.cnf` fájlt ezzel a tartalommal:

```
[mysqladmin]
user = root
password = új-jelszó
```

- ha esetleg nem *root*-ként készítettük volna a fájlt:

```
chown root:root /root/.my.cnf
```

- Mindenképp: `chmod 0600 /root/.my.cnf`

Ezzel elértük, hogy titkos jelszavat adtunk az adatbázis rendszergazdának és még a szerver is jól fog funkcionálni (különben induláskor és leálláskor is jelszót kérne, ami nem lenne jó, ha nem is vagyunk gépközelben). Ezt a jelszót adjuk át az adatbázis-rendszergazdának is, hogy egyáltalán tudjon valamit kezdeni a rendszerével és létrehozassa a felhasználóit.

Ahhoz, hogy a **mysql** hibaüzeneti magyarul jelenjenek meg, keressük meg ezt a sort és módosítsuk az `/etc/mysql/my.cnf` fájlban

```
language=/usr/share/mysql/hungarian
```

Rengeteg apró trükk is számba jöhet, mely mind a kényelmünket szolgálja. Például sokkal használhatóbb a héj, ha a héj „kérdése” így néz ki:
[webgazda@alfa:/usr/doc/] \$ Ezt úgy érhetjük el, ha a saját `.bashrc`-nkbe ezt írjuk be: `PS1='[\u@\h:\w]\$'`

Egy minta `.bashrc`:

```
set meta-flag on                # ezek a magyar bill. kezeléshez szükségesek
set convert-meta off
set output-meta on
PS1='[\u@\h:\w]\$'              # ez állítja be a shell prompt-ot
#\u az felhasználó neve, \h a gépnév, \w pedig az aktuális könyvtár
export PS1                      # ezzel pedig közzétesszük azt magunknak
umask 027                      # állítsuk be az umask értékét
# ez annyit tesz, mint umask -S u=rwx,g=rx,o=
```

¹²⁹ A MySQL-ben lévő felhasználók nem egyeznek a UNIX rendszerünkben lévő felhasználókkal, azokat külön kell karbantartani.

¹³⁰ Bővebben az `/usr/share/doc/mysql-doc` könyvtárban találhatunk információkat erről, kifejezetten a 6. fejezetben.


```
export LS_OPTIONS='--color=auto -h' # kellemes színeket kapunk az ls parancshoz
eval `dircolors`
alias ls='ls $LS_OPTIONS'           # ezzel érvényesítjük
alias ll='ls $LS_OPTIONS -l'        # legyen egy-két rövidebb parancs a hosszú
alias l='ls $LS_OPTIONS -lA'        # paraméterezés helyett
HISTFILESIZE=0                      # ez biztonsági okok miatt hasznos, ekkor nem
                                    # olvasható el, hogy milyen parancsokat használtunk eddig
EDITOR=/usr/bin/joe                 # ha nem adjuk meg, az alap szövegszerkesztő a "vi"
```

Ha ezt az `/etc/skel` könyvtárban helyezzük el, akkor minden új létrehozott felhasználónak ez kerül be a könyvtárába.

Készítsünk egy `/etc/environment` fájlt a következő tartalommal: `LANG=hu_HU`

Ezután, amikor lehet, magyarul fognak megjelenni az üzenetek. Mivel ahhoz, hogy ez érvénybe lépjen, újra be kellene jelentkezni, ezért írjuk be ezt a parancssorba: `export LANG=hu_HU`.

Magyar ékezetes kiosztásra az angolról `loadkeys hu` paranccsal válthatunk. Értelemszerűen visszaváltani angolra a `loadkeys us` –el lehet. (Szükségünk lehet a `fonty` csomagra is.)

A hat konzol között `ctrl-alt-F1..6` gombokkal válthatunk.

2.2 Az `inetd.conf` finomhangolása – a nem biztonságos szolgáltatások letiltása

Az `inetd` program az Internetes Szuperszerver. Feladata az, hogy a ritkán használt és / vagy speciális szolgáltatásokat elindítsa, ha kívülről kérés érkezik valamelyik meghatározott portra. Mivel az Apache (és a többi itt futó szolgáltatás is) „*Standalone*”¹³¹ módban fut, ezért esetünkben nincs szükség az `inetd` munkájára.

Igazság szerint a jelen felállásban nincs szükség egyetlen szolgáltatásra sem azok közül, melyeket az `inetd` nyújt. Ezért az egész `inetd` programot letilthatjuk. Sajnos ez a program és a hozzá tartozó fájlok sok más jóval együtt a `netbase` csomagban vannak – így nem törölhetjük le őket a rendszerből. Amennyiben mégis szükségünk lenne néhány szolgáltatásra, mely az `inetd`-ből futna, használjunk egy profibb helyettesítő csomagot, mint amilyen pl. az `xinetd`, vagy `rlinetd`.

Először is állítsuk le az `/etc/init.d/inetd stop` paranccsal. Ezután szerkesszük át az `/etc/inetd.conf` fájlt úgy, hogy minden bejegyzés elé tegyünk egy „#” jelet. Erre azért van szükség, hogyha valaki el is indítaná az `inetd` programot, akkor se tudjon vele mit kezdeni. Ezután írjunk az `/etc/init.d/inetd` fájl első sorába egy „`exit 0`” sort. Ekkor a inicializáló script indítás helyett kilép. Az `/etc/inetd.conf`

¹³¹ Vagyis önálló üzem mód.

jogosultságait állítsuk a következőre: `chmod u+r,u-w,a-wrx inetd.conf`¹³² Ezek után biztos ami biztos alapon adjunk neki *immutable bit*-et: `chattr +i inetd.conf`. (Sőt, akár `chmod a-x /usr/sbin/inetd`, de ez könnyen kizárható.)

Hogy a paranoia teljes legyen, állítsuk be a `tcp_wrapper`¹³³ fájljait is: `/etc/hosts.deny`-ban kommentezzünk ki mindent és tegyük be az `ALL:ALL` és az `ALL: PARANOID` sorokat. Az első sor minden olyan címről érkező kérést megtagad, amelyik nincs benne a `hosts.allow` fájlban. A másik szabály az olyan gépekre vonatkozik, melyeknek az IP címe nem egyezik meg a nevük szerint a DNS-ből, vagy az `/etc/hosts` fájlból lehívott név-IP cím hozzárendelésnek.

A *Standalone* módban futó szervizeknek a saját konfigurációs állományaikban lehet meghatározni, hogy milyen tartományokat szolgáljanak ki és milyeneket nem. Ezekre az adott helyen kitérek.

A következő felesleges szolgáltatás a `portmap`. Ez a démon szolgáltatná a SUN féle RPC-k listáját¹³⁴ (*Remote Procedure Call*, Távoli Eljárás Hívás). Mivel ezeket a szolgáltatásokat is az `inetd` indítja – és azt már megszüntettük – erre sincs szükség. Állítsuk le a szolgáltatást az `/etc/init.d/portmap stop` paranccsal, majd írjunk egy `exit 0` sort az `/etc/init.d/portmap` script első sorába.

2.3 A levelező démon beállítása

A nagyfokú biztonságossága miatt a `postfix` csomagot választottam. Ezt az IBM-nél kezdték fejleszteni. Szerzője *Wietse Venema*, aki többek között a `tcp_wrapper` program írója is és biztonsági problémák megoldásán dolgozik. A Postfix-et már a tervezési fázisban a biztonságra hangolták. A program neve arra utal, hogy az elterjedt és viszonylag könnyen feltörhető `sendmail` program lecserélésével utólagos biztonsági foltozást kapunk. Mivel ez egy Web és nem egy levelező szerver lesz, nincs szükségünk a `sendmail` extra funkcióira. Nincs szükségünk a `qmail` gyorsaságára és teljesítményére sem. A Postfix ezért kitűnő választás.¹³⁵

A Postfix a Potato-ban két változatban is megtalálható. A `postfix` csomag a standard programot tartalmazza, míg a `postfix-tls` csomag (mely a *non-US* szekcióban van) egy TLS (*Transportation Layer Security*, az RFC2246 szerint) bővítéssel rendelkezik.

¹³² Ez megegyezik a `chmod 0400-val`

¹³³ Ez a program végzi az `inetd` által nyújtott szolgáltatások IP cím vagy gépnév szerinti szűrését. A `hosts.allow` fájlba írjuk be azokat a gépeket, melyeket ki akarunk szolgálni az `inetd`-n keresztül, a `hosts.deny`-ba pedig azokat, melyeket semmiképp sem.

¹³⁴ Ez listázza a kliensek számára azt, hogy milyen RPC típusú szolgáltatások vannak a gépen és azok mely DARPA porton futnak.. Bővebben: `man portmap`.

¹³⁵ Az `/usr/doc/postfix-tls/COMPATIBILITY` fájlban részletesen le van írva, hogy mely funkciók `sendmail`-kompatibilisek.

Ez az OpenSSL rendszerkönyvtárat használja. Az RFC2487¹³⁶ leírja a TLS SMTP-be való integrálását. Ezek segítségével készítette el *Lutz Jänicke* a TLS bővítést a Postfix-hez. Segítségével a következők érhetők el, amennyiben mindkét gép, melyek között a kapcsolat folyik, ismeri ezt a protokollt:¹³⁷

- Titkosított e-mail szállítás a gépek között
- A fogadó gép beazonosítása, hogy azonos-e azzal, akinek mondja magát.
- A küldő gép beazonosítása, levéltovábbítás céljából (*relaying*)

És amire nem képes:

- A felhasználók leveleinek biztonságát megőrizni - ez a titkosítás csak szállítási rétegre vonatkozik
- A küldő azonosítása

A fentiek eléréséhez használjunk `gpg`-t vagy `pgp`-t

Bár még nagyon kevés gépen van ilyen protokollt ismerő levelező démon¹³⁸, mégis ajánlom ennek a használatát. Később egyre jobban el fog terjedni, és addig is legalább a saját szervereink legyenek nagyobb biztonságban.

Konfigurációs állományait az `/etc/postfix` könyvtárban találjuk. A fő beállításokat a `main.cf` fájl tartalmazza. Nyissuk meg kedvenc szerkesztőnkben és módosítsuk a következő sorokat:

```
myhostname = alfa.boresszormegyar.hu # a saját gépünk neve
mydomain = boresszormegyar.hu        # mi a bejegyzett domén-nevünk
# myorigin = $mydomain # ha ez lenne a fő mail szerver, akkor ezt adnánk forráscímnek
# milyen gépeknek vagyunk a levelező szervere: ( „A” típusú DNS rekord kell)
mydestination = $myhostname, localhost.$mydomain, www.$mydomain
```

Ha szeretnénk a felhasználóknak *Családnév.Keresztnév* stílusú címzést is biztosítani, szűrjük be a következő sort is:

```
canonical_maps = hash:/etc/postfix/canonical
```

Alapbeállításként ez a fájl a manuál oldal szövegét tartalmazza, ezért töröljük ki / le. Majd szerkesszük meg a hivatkozott fájlt a következőképp:¹³⁹

```
# user Név.Név
rgazda Kis.Jozsef
webgazda Nagy.Istvan
abgazda Kovacs.Lajos
#stb.
```

Ha virtuális doméneket is akarunk használni, akkor szűrjük be ezt a `main.cf`-be:

```
virtual_maps = hash:/etc/postfix/virtual
```

Ezután szerkesszük a hivatkozott fájlt:¹⁴⁰

¹³⁶ Mindkét RFC szövege megtalálható az `/usr/doc/postfix-tls/tls/html` könyvtárban.

¹³⁷ Pl. a Netscape Mail 4.5-ös verziójától kezdve ismeri ezt a szabványt, így ezzel küldhetünk a szerverünk felé titkosítva is.

¹³⁸ Pl. QMailTLS és Zmailer-crypt

¹³⁹ Bővebb információért: man 5 canonical

```
# ezeket a doméneket is bejegyeztük, ezért nekik is fogadjuk a leveleket.
borgyar.hu
szormegyar.hu
```

Ha kérietlen levelek szerverszintű szűrésére van szükségünk, olvassuk el a `/usr/doc/postfix-tls/html/uce.html` fájlt. Fontos, hogy ne használjunk „Open Relay”-t, vagyis ne továbbítsuk bárki leveleit átjátszóként, mert ekkor a *spamer*¹⁴¹-ek céltáblájává válhatunk, ráadásul bekerülünk az ORBS¹⁴²-be, és akkor sok helyre levelet sem küldhetünk majd.

Mivel ez nem egy levelező szervernek lett szánva, limitáljuk az egyszerre futható levélküldések számát az `/etc/postfix/master.cf`-ben 50-ről pl. 10-re¹⁴³.

```
# =====
# service type private unpriv chroot wakeup maxproc command + args
#          (yes)   (yes)   (yes)   (never) (50)
# =====
smtp      inet  n       -       n       -       10      smtpd
```

Nyissuk meg az `/etc/aliases`¹⁴⁴ fájlt is és módosítsuk így:

```
# a postmasternek szánt leveleket a rendszergazda kapja meg két helyre is
postmaster: rgazda@gyakranolvasom.hu root
# a webmasternek szánt leveleket ketten is megkaphatják
webmaster: rgazda@gyakranolvasom.hu webgazda, wgazda@gyakranolvassa.hu
# minden root-nak szánt levelet továbbítsunk
root: rgazda@gyakranolvasom.hu
info: titkarno
```

Indítsuk újra a levelező szervert : `/etc/init.d/postfix restart`

Ha valamit elgépeltünk a fájlokban, akkor most lehet, hogy nem fut a démon. Győződjünk meg erről így: `ps aux | grep postfix` Ekkor három programnak kell megjelennie: `master`, `pickup`, `qmgr`. Ha ezek láthatóak a sorok között, akkor minden rendben van.

A titkosítás beállítása

A következőkre van szükség a titkosításhoz:

- 1 db titkos kulcs a szerverhez
- 1 db nyilvános kulcs a szerverhez, melyet egy CA¹⁴⁵ hitelesített (~digitálisan aláírt), mely azt bizonyítja, hogy cégünké az adott gépnév (és domén).

¹⁴⁰ man 5 virtual

¹⁴¹ Kérietlen, főleg üzleti hirdetést tartalmazó levelek küldői.

¹⁴² <http://www.orbs.org>, az Open Relay-es szervereket tartalmazó adatbázis.

¹⁴³ Kisebb (régőbbi) hardveren ez akár 2 is lehet (legyen).

¹⁴⁴ Ez tartalmazza azt, hogy ki kicsoda, és az adott felhasználói számlára érkező leveleket kell-e továbbítani.

¹⁴⁵ Certificate Authority, vagyis kb. azonosító hatóság, vagy digitális közjegyzőség.

- 1 db CA igazolás a CA nyilvános kulcsával

Ei kell készítenünk tehát egy igazolást arról, hogy azok vagyunk, akik vagyunk (vagy vásárolhatunk egyet egy erre szakosodott cégtől, de ez még felesleges). A következő lépéseket kell megtenni:

0. Megkeresünk egy segédprogramot és beállítjuk azt igényeinkhez.
1. Legyártunk magunknak egy saját CA kulcsot, hogy CA-vá válhassunk.
2. Legyártunk a levelező szerver számára egy szerver kulcsot.
3. Ezt aláíratjuk a CA kulccsal.
4. Bemásoljuk a fájlokat a helyükre.
5. Megszerkesztjük a konfigurációs fájlt
6. Újraindítjuk a levelezőt.

Részletesen:

- (0) A dokumentáció¹⁴⁶ elolvasása után keressük meg a következő fájlt: `/usr/lib/ssl/misc/CA.pl` Másoljuk át a `/root` könyvtárba és hívjuk be kedvenc szövegszerkesztőnkbe. Az alábbi minta alapján szerkesszük át a fájlt (csupán egy `-nodes` paramétert kell beírunk két helyre¹⁴⁷):

```
[...]
# create a certificate
system ("$REQ -new -x509 -nodes -keyout newreq.pem -out newreq.pem $DAYS");
$RET=$?;
print "Certificate (and private key) is in newreq.pem\n"
} elsif (/^-newreq$/) {
    # create a certificate request
    system ("$REQ -new -nodes -keyout newreq.pem -out newreq.pem $DAYS");
[...]
```

Erre azért van szükség, mert az indításkor nem szabad, hogy a kulcs fájl kódolt legyen, különben a `postfix` nem fogja tudni beolvasni. Keressük meg továbbá a `$DAYS=' - days 365'` sort. Ennyi nap után jár le az igazolás, évente meg kell újítanunk saját magunknak. Sajnos nem jöttem rá, hogyan lehetne ezt működőképesen megnövelni.

- (1) Most futtassuk ezt a programot: `./CA.pl -newca` Ez a kis programocska legyártja nekünk a megfelelő igazolást (CA). Először is adjunk meg neki egy jelszót kétszer (erre később emlékeznünk kell!). Majd – az Apache-SSL-hez hasonlóan – töltsük ki itt is az azonosítói adatokat. Ekkor már mi vagyunk a saját azonosító-szolgáltatónk (CA).
- (2) A következő lépésben futtassuk a `./CA.pl -newreq -t`, és töltsük ki ezt is hasonlóképpen:

¹⁴⁶ `/usr/doc/postfix-tls/tls/html/index.html` és kifejezetten a `certificates.html`

¹⁴⁷ A Postfix-TLS Debian csomag felelőse már megjegyezte, hogy ez a megváltoztatott `CA.pl` benne lesz az új kiadásban.

14. kép - Postfix – igazolás készítése a CA.pl programmal

A *challenge password* és *optional company name* mezőket üresen is hagyhatjuk. (Ezzel elkészítettünk egy igazolást).

(3) Ha ez kész, futtassuk le ezt: `./CA.pl -sign` Írjuk be a jelszót, a felmerülő kérdésekre válaszoljunk Igen-nel. (Ezzel aláírtuk az igazolást.)

(4) Másoljuk be a fájlokat a helyükre :

```
cp /root/demoCA/cacert.pem /etc/postfix/CA.pem
cp /root/newcert.pem /etc/postfix/igazolas.pem
cp /root/newreq.pem /etc/postfix/privatkulcs.pem
```

(5) Szerkesszük meg a `main.cf`-et:

```
smtp_tls_key_file = /etc/postfix/privatkulcs.pem # a titkos kulcsunk
smtp_tls_cert_file = /etc/postfix/igazolas.pem # a nyilvános igazolás, kik vagyunk
smtp_tls_CAfile = /etc/postfix/CA.pem # a CA igazolása (ezt is mi csináltuk most)
smtpd_tls_key_file = /etc/postfix/privatkulcs.pem # a "d" re végződő paraméterek
smtpd_tls_cert_file = /etc/postfix/igazolas.pem # segítségével kliens is lehet
smtpd_tls_CAfile = /etc/postfix/CA.pem # a szerverünk és így is lehet azonosítani
```

A titkos kulcsot csak a rendszergazdának szabad látnia:

```
chown root /etc/postfix/privatkulcs.pem
chmod 400 /etc/postfix/privatkulcs.pem
```

(6) Indítsuk újra a levelezőt: `/etc/init.d/postfix restart`

Ha valamit elgépeltünk a fájlokban, akkor most lehet, hogy nem fut a démon. Győződjünk meg erről így: `ps aux | grep postfix` Ekkor három programnak kell megjelennie: `master`, `pickup`, `qmgr`. Ha ezek láthatóak a sorok között, akkor minden rendben van.

A többi alapbeállítás általában megfelelő minden esetre. Az egész programot már a tervezéskor úgy hangolták, hogy biztonságos legyen.

Ha minden jól ment, akkor a levelező szerverünk rendesen be lett állítva igényeinkhez. Mindenképp olvassuk végig a dokumentációt, amint van rá egy kis időnk. (pl. `lynx /usr/doc/postfix-tls/html/index.html`)

Linkek: <http://www.postfix.org> , <http://www.aet.tu-cottbus.de/personen/jaenicke>

2.3 Másik gép használata a fordításokhoz, miért?

Mivel éles szerveren biztonsági okokból nem tarthatunk semmilyen fejlesztő-eszközt, fordítót, stb., ezért nagyon ajánlott egy másik – rendszergazdai adminisztrációs gépet fenntartani. Ha fordítókat tartanánk a szerveren, bármelyik felhasználó – vagy épp a betörő forráskódból fordíthatna magának a gépen programokat. Ez kerülendő.

A szervereken üzembe helyezés után általában se billentyűzet (BIOS beállítás!), se egér, se monitor nem szokott lenni. A szervert mindig a hálózatról, egy másik gépről – a rendszergazda gépéről lehet menedzselni.

Jelöljük ki (vagy vásároljunk) egy gépet a rendszergazda számára. Ez lehetőleg megfelelően jó tudású / teljesítményű gép legyen. Az se baj, ha ez egy megfelelő eszközökkel (hálózati kártya, modem) ellátott laptop. Ekkor a rendszergazda azt mindenhol magával viheti, és szükség esetén nála van minden fontos és kézreálló eszköz.

Továbbá ezen a gépen kell kipróbálni először az új programokat, kernelverziókat, hogy az éles rendszerekre már tesztelve, megismerten kerülhessenek fel az újítások.

Ez a gép legyen elzárva a hálózat többi részétől – lévén ez a rendszer nem éles, tehát nincsenek (ne legyenek) publikus szolgáltatásai, hiszen fejlesztői rendszer, ezért érzékeny lehet a támadásokra. Ez a rendszer mindig érzékeny adatokat és programokat tartalmazhat. Ezért nagyon körültekintőeknek kell lennünk. Ebből a rendszerből csak kifelé lehessen látni, kintről befele ne.

2.4 Személyre szabott kernel konfigurálása és fordítása kézzel és a „kernel-package” csomaggal. A „lilo” beállításai.

Ez egy nagyon tág és mély témakör. *II. Alapfogalmak / 2. A Linux kernel* c. fejezetben már kitértem a kernel sajátosságaira, azokat itt nem szeretném elismételni. Szintén nem térhetek ki a különböző hardver eszközök kiválasztására és beállítására. Azokat a lényeges pontokat viszont megpróbálom érinteni, melyek a biztonság szempontjából fontosak, és / vagy mindenképp bele kell kerülnie a kernelbe.

A 2.2-es linux konfigurálásához van egy részletes magyar nyelvű útmutató, mely itt [7] található. Igaz ez már egy kicsit idejétmúlt, a 2.2.4-es kernelt taglalja, viszont 25 oldal terjedelmével elég kimerítő. Ez a link¹⁴⁸ pedig a magyar nyelvű *linux-kernel-HOGYAN*-t fedi, igaz, ez is elég idejétmúlt már, viszont jó elméleti kiindulópont lehet.

¹⁴⁸ <http://www.vmg.sulinet.hu/vmg/home/szamtech/linux/node233.htm> A Linux Kernel HOGYAN (Brian Ward, bri@blah.math.tu-graz.ac.at) 1997. május 26. Verzió: 0.80 fordítása alapján készült.

Olvassuk végig először az eljárások menetét és csak utána kezdjük hozzá a tényleges gyakorlati megvalósításhoz. Én a `kernel-package` csomag segítségével való fordítást ajánlom, de meg kell ismernünk a kézzel való fordítás menetét is.

Szerezzünk be egy kernel forráskódot, ha lehet, akkor pl. az `apt-get install kernel-source-2.2.16` paranccsal. Ha ennél frissebb vagy speciálisabb forrást szeretnénk, akkor valamelyik kernel tükörről beszerezhetjük azt. Ha szükség van kernelszintű fájlrendszer-titkosításra, akkor a nemzetközi változat „feltjét” is töltsük le. Továbbá szerezzünk be minden olyan forráskódot, amelyre az adott hardverhez szükségünk lehet, de még nincs benne a hivatalos kernelben (pl. alaplapi CPU hőmérőket és ventilátor fordulatszám-mérőt kezelő programrész, az *lm_sensors* nevű folt.¹⁴⁹). Ha megvan minden és a fordításhoz szükséges bináris csomagok is fenn vannak a gépen (`gcc`, `binutils`, `make`, `libc6-dev`, `bin86`, ha kell `menuconfig`: `ncursesX.X-dev`, `make xconfig`: `tkX.X-dev`, azok a csomagok, melyeket ezek igényelnek, `gzip`, `grep`, `shellutils`, stb.).

Ha i586, vagy i686 architektúrájú processzorral rendelkezünk a szervergépben, akkor a fejlesztői (rendszergazdai) gépre tegyük fel a `pentium-builder` csomagot. Segítségével a célgép processzorához tudjuk optimalizálni a kernel (és minden más lefordított C program) kódját, mely viszonylag nagy sebességnövekedést eredményezhet. Ha a program fent van, adjuk a következő parancsot: `export DEBIAN_BUILDARCH=pentium` (i686 esetén pedig `pentiumpro`). Ha nem akarjuk ezt minden fordítás előtt begépelni, tegyük be a `~/.bashrc` fájlunkba. Ezután az összes C program kódja a kért processzorhoz lesz optimalizálva. Ekkor az ennél kisebb processzorokon az adott program / kernel nem fog futni (pl. i486).

A kész lefordított kernel a következő részekből áll:

<i>fájl(ok)</i>	<i>méret</i>	<i>forma</i>	<i>funkció</i>
<code>arch/i386/boot/bzImage</code>	~600kb	bináris	A rendszermag.
modulok	1-2Mb	bináris	az induláshoz nem szükséges programrészek
<code>./System.map</code> ,	~300kb	szöveges	A rendszerhívások listája
<code>./.config</code> ,	~10kb	szöveges	A kernel konfigurációja

9. táblázat - A kernel részei

A rendszermag elkészülte után le lesz tömörítve és a fenti könyvtárba lesz másolva. Minden olyan kódrészt, amely a rendszer indulásához nélkülözhetetlen bele kell fordítani a kernelbe és nem szabad modulba tenni (pl. ha IDE vezérlős lemezünk van és arról indul a rendszer, akkor az IDE vezérlő támogatást mindenképp fordítsuk be).

¹⁴⁹ <http://www.lm-sensors.nu>

Bizonyos kódrészeket egyáltalán nem is lehet modulba fordítani. A modulokat később pl. a `modprobe` vagy `insmod` paranccsal lehet betölteni, az `rmmod`-al kiszedni és az `lsmod`-al listázni (`modutils` csomag).

A `System.map` fájl a kernel rendszerhívásait és szimbólumait tartalmazza. Induláskor ezt a kernel (`klogd`) beolvassa. Ez megkönnyítheti a kernel üzeneteinek megértését, ugyanis ekkor emberi (angol) nyelven szól hozzánk.

A `.config` fájlt jó eltenni későbbre, hogyha egy kis változás miatt újra kell fordítani, akkor ne kelljen előről kezdeni a „konfigolást”.

A kézzel való fordítás lépései:

1. Csomagoljuk ki a kernelforrást, pl. az `/usr/src` könyvtár alá:

```
tar -xzf linux-2.2.16.tar.gz; cd linux
```

2. Ha kell, foltozzuk be a szükséges plusz kódokkal. Pl.:

```
cat folt.txt | patch -p1 -E -N -s -d /usr/src/linux
```

Ha úgy döntünk, alkalmazzuk az OpenWall-féle kódot, csomagoljuk ki:

```
tar -xzf linux-2.2.16-owl.tar.gz
```

Majd pedig foltozzunk:

```
cat linux-2.2.16-owl/linux-2.2.16-owl.diff | patch -p1 -E -N -s -d /usr/src/linux
```

Ha használni akarjuk a LIDS rendszert is, olvassuk el a függelékben lévő útmutatót.

3. Mivel már tudjuk, hogy melyik hardver elemet milyen vezérlővel fogunk használni, ezért keressük elő ezt a jegyzetünket. Végezzük el a kernel konfigurálását. Ezt háromféleképpen is megtehetjük: `make config`, `make menuconfig`, `make xconfig` parancsokkal. Az első elég kényelmetlen, mert egyenként rákérdez minden egyes lehetőségre. A második már jó, ekkor egy színes, menüs, de még szöveges programocska fut, melyből kényelmesen kiválogathatjuk, hogy mi kell nekünk. Ehhez szükséges az `ncurses-dev` csomag, mivel ez a programocska is le kell, hogy forduljon és `ncurses` képernyőkezelő rutinokat használ. Kérésre angol nyelvű rövid leírást kapunk minden egyes opcióról.

15. kép - make menuconfig

A harmadik változat a legkényelmesebb, egy TCL/Tk függvénykönyvtárat használó grafikus X11 felületű beállító-programot kapunk.

16. kép - make xconfig

Akkor nézzük a fontosabb beállításokat. Én a `make xconfig` felületet használom. Mi az ami mindenképp benne kell, hogy legyen és mi az ami semmiképp se kell a konfigurunkban:

- *Code maturity level options / Prompt for development and/or incomplete code/drivers*: a félkész/fejlesztés alatt lévő eszközvezérlők és funkciók megjelenítése választási lehetőségként. Kell.
- *Processor type and features / Processor family*: válasszuk ki, hogy milyen processzor van a gépünkben. *Math emulation* – ez semmiképp sem kell, ha van FPU¹⁵⁰ a gépünkben. *MTRR*¹⁵¹ *support*: ha i686-os processzorunk van akkor ennek a használata gyorsítja a végrehajtást, kellhet. *SMP support*: ha egynél több processzor van a gépünkben, akkor kell.
- *Loadable module support / Enable ...*: kell, hiszen szeretnénk modulokat használni. *Kernel module loader*: mindenképp kell – kernelszintű modul betöltő.
- *General setup / Networking support*: [y], hiszen épp hálózatra tesszük a gépet. Itt be lehet állítani a PCI buszt, ha van tegyük. A *System V IPC*, a *BSD process Accounting*, *Sysctl support*, *kernel support for ELF bin.*, mindenképp kell. Ha akarjuk használni a párhuzamos portot, tegyük be, de ez is egy betörési lehetőség lehet a gépre, igaz helyi. Én javaslok az *APM*¹⁵² *support* kernelbefordítását és megfelelő beállítását (SMP módban nem megy). Egyrészt áramot és ezzel pénzt spórolhatunk meg, másrészt megnöveljük a szerver élettartamát, mivel kevesebbet „pörög” a gép processzora. Figyelem! A *suspend* funkciót tiltsuk le a BIOS-ban, csak a *doze mode*-ot engedélyezzük (esetleg a *stand by* módot

¹⁵⁰ Floating Point Unit, vagy matematikai társprocesszor.

¹⁵¹ Memory Type Range Register

¹⁵² Advanced Power Management: Energiatakarékos üzemmódra kapcsolja a gépet, ha az nincs használva.

is), mely – ha nincs szükség rá – le szabályozza a processzor frekvenciáját.¹⁵³ A *suspend* funkció hatására a gép alvó-üzemmódra kapcsol és csak pl. a billentyűzet megnyomásával ébreszthető fel – ez szervereken nem előnyös. Ha a gép nagyon nagy forgalmat bonyolít, akkor az egész APM-et hagyjuk ki. Ha viszont keveset forgalmaz, akkor fontoljuk meg használatát. Javasolom továbbá az RTC¹⁵⁴ GMT¹⁵⁵-hez való állítását is.

- *Plug and Play support*: ha van ilyen kártya a gépben, akkor kapcsoljuk be.
- *Block devices*: válasszuk ki, milyen blokkos eszközöket (floppy, merevlemez) akarunk használni. Azt az eszközt, amiről a rendszer indul, fordítsuk kernelbe, a többi mehet modulba is. Ha nincs szükség rá, a floppy-t ki is hagyhatjuk. Ha alaplapi IDE vezérlő chipset-ünk van és a Linux támogatja ezt, akkor válasszuk azt is¹⁵⁶. *Loopback Device support*, *Network block device support* kell. Ha valamilyen RAID, vagy párhuzamos portra csatlakoztatható IDE eszközünk van, válasszuk ki.
- *Networking options*: itt ki kell választanunk, hogy milyen hálózati funkciókra lesz szükség. *Packet socket*: [m], *K./U. netlink socket*: [y], *Network firewalls*: [y], *Unix domain sockets*: [y], *Socket filtering*: [y], *TCP/IP networking*: [y], *IP firewalling*: [y], *IP firewall packet netlink dev.*: [y] Ezek többek között a tűzfal és csomagszűrő funkciók. *Masq* és *proxy* nem kell, mert ennek a szervernek nem az a feladata. *IP aliasing*: [y] – ez kell, ha IP-s *virtualhost*-ot szeretnénk. *TCP syncookie support*: [y] – ez a DoS típusú támadásoktól véd. *Allow large windows*: [y] Ha szükségünk van egyéb hálózati protokollokra, funkciókra, akkor jelöljük ki azokat is.
- *SCSI support*: ha van ilyen eszközünk, válasszuk ki. (pl. merevlemez)
- *Network device support*: [y], *Dummy*: [y], Ha van pl. üvegszál hálózati eszközünk, akkor válasszuk ki. Ha van *ARCnet*, *Ethernet*, *Appletalk*, *Token ring*, *WAN*, *Amateur Radio* hálózati eszközünk válasszuk a megfelelő vezérlőt.
- Nem hiszem, hogy a szerverben infravörös, ISDN, és / vagy régi kártyás CD-ROM eszközeink lennének. Ezeket kihagyhatjuk.
- *Character devices / Virtual terminal*: [y], *Support for console...*: [y], ha kell, soros port vezérlőnk is válasszuk ki. *Unix98 PTY support*: [y], *Watchdog support*: [y], (lesz majd szoftveres) *Enhanced Real Time Clock Support*: [y] Egérre nem hiszem, hogy itt szükség lesz. Nyomtatóra, botkormányra, képfeldolgozó eszközre sem.
- *Watchdog cards: / software watchdog*: [m], ezt később tárgyalom.
- Ha van *Ftape* (floppy portra kapcsolható szalagos egység), akkor azt is.

¹⁵³ Különösen javasolt, ha nincs légkondicionáló és a helyiségben és nyáron a levegő 30 hőmérséklete Celsius fok fölé emelkedik a gépteremben. Továbbá hasznos akkor is, ha a processzor túl van hajtva (overclocking) – bár ez szervereken egyáltalán nem javasolt.

¹⁵⁴ Real Time Clock: valós idejű óra a számítógépben.

¹⁵⁵ Greenwich Mean Time: az egységes csillagászati földi idő, a 0-s időzóna.

¹⁵⁶ Ha esetleg nem találunk, akkor töltsük le az IDE foltot Hedrick könyvtárából, ez a legfrissebb fejlesztésű IDE vezérlőket tartalmazza. [ftp://ftp.hu.kernel.org/pub/linux/kernel/people/hedrick](http://ftp.hu.kernel.org/pub/linux/kernel/people/hedrick)

- *Filesystems / Second extended fs support*: [y] – erről fogunk ui. indulni. */proc...*, */dev/pts...*: [y] Itt még sok egyéb választásunk is van. Ha esetleg más operációs rendszereket is tartanánk ugyanazon a gépen (a mi esetünkben nem), akkor válasszuk ki, ami kell.
- *Network file systems / NFS fs supp.*: [m] ez a klienshez kell, szervert meg nem tanácsos üzemeltetnünk, az kimarad. Ha van a környezetben más gyártótól rendszerszoftver és szükség is van rá, hogy rendszerünk lássa azokat, akkor válasszuk ki ami kell.
- *Partition types*: ha ugyanazon a gépen más op'rendszereket is tartunk, jól jöhet a nekik megfelelő speciális partíciós tábla vezérlője, egyébként semmi szükség rájuk.
- *Native language support / NLS ISO 8859-1 és -2 modulba*.
- *Console drivers / VGA text console*: [y], - ha VGA kártya van a gépben. Támogathatjuk továbbá az *MDA* egyszínű monitorokat is. Szerverben a *Frame buffer*-nek nem sok értelme van, úgyszintse rajta monitor. Ha nem választunk itt ki semmit, akkor nem jutunk be a gépbe, csak a soros porton!
- *Sound* – itt hangkártyának semmi értelme, nem kell.
- Ha betettük az *OpenWall*-féle foltot, akkor itt van egy *Security Options* menü is, ami alatt az összes kérdésre válaszoljunk yes-el.
- Ha a LIDS rendszert is befoltoztuk, olvassuk el a részletesebb útmutatót a Függelékben!
- *Kernel hacking – Magic SysRq key* : [y] – ha a rendszer lefagyna, akkor az *alt-printscreen* billentyűvel még esetleg feléleszthetjük a konzolt, vagy legalább kiíratjuk a veremtárat és a gyorsítótárakat. (Olvassuk el a doksiját!)

Miután mindent beállítottunk mentsük el a változásokat és lépünk ki.

4. Ebben a lépésben a fordító a `.config` fájl alapján beállítja a függőségeket – csak az fog lefordulni a forrásból, amire szükség van a mi beállításainkhoz. Ezt a `make depend` paranccsal tehetjük meg. Ezt a lépést nem lehet kihagyni!

5. Most elindítjuk a kernel fordítását: `make -j2 bzImage`¹⁵⁷

6. Ha nem kapunk hibaüzenetet, akkor indítsuk a modulok fordítását.

```
make -j2 modules
```

7. Ha itt se kaptunk hibaüzenetet, akkor a `make modules_install` paranccsal felteszi a `/lib/modules/kernelverzió` könyvtárba a kész modulokat. Ha kész mozgassuk át őket a szerverre.

8. Másoljuk a magot, a térképet és a konfigot a `/boot`-ba

```
scp arch/i386/boot/bzImage root@alfa:/boot/vmlinuz-kernelverzió
scp System.map root@alfa:/boot/System.map-kernelverzió
scp .config root@alfa:/boot/config-kernelverzió-gépnev
```

¹⁵⁷ A `make` parancsot a `-j` –vel lehet paraméterezni, hogy hány konkurens folyamatot indítson a jobserver. Egy i686 processzor esetében javaslok a `-j2` opciót. Ahány processzorunk van a gépben, annyszor kettő. Ez nagyban meggyorsítja a fordítást.

9. (Innentől minden a szerveren történik) Szerkesszük meg az `/etc/lilo.conf` fájlt, hogy az új kernelünk felkerülhessen az indítható kernelek közé. Ha kész futtassuk a `lilo`-t. Itt látható egy mintapélda:

```
boot=/dev/hda          # az eszköz, ahova a boot menedzser települjön
install=/boot/boot.b   # ez az i386 architektúra miatt szükséges valós módú kód
map=/boot/map          # és lemeztérképe
vga=5                  # milyen VGA szöveges módba váltson a konzol, itt 80x34
delay=50               # várakozás tizedmásodpercben, mielőtt a kernelt indítja.
prompt                 # mindenképp jelezze ki a választómenüt
timeout=50             # ha nekikezdünk gépelni, de abbahagyjuk...
default=sajat          # melyik kernel induljon el, ha nem választunk
message=/boot/message  # a lilo prompt elé írhatunk üzenetet.
root=/dev/hda3          # hol van a gyökér fájlrendszer
read-only              # csak olvasható módban fűzze fel először a rendszert

#a gyári kernel mindig legyen fenn, hátha kell.
image=/vmlinuz          # a kernel helye
password=jelszo         # mi legyen a jelszó, arra az esetre
restricted             # ha paraméterek akarunk átadni a kernelnek induláskor158
label=gyari             # mi legyen a kernel neve a lilo prompt-nál

image=/boot/vmlinuz-2.2.16
password=mehet
restricted
label=sajat
append="idebus=45"      # paraméterezhetjük az egyes kódrészeket az append
                        # parancs segítségével. (opcionális)
```

Ezek után, ha minden hiba nélkül futott le, újraindíthatjuk a szerveret. Mindenképp legyen egy stabil, működő kernel a `lilo.conf`-ban, nehogy kizárjuk magunkat a rendszerből. Érdeemes minden jól működő és használatban lévő kernelből *boot* floppy-t is gyártani, hátha megsérül az MBR.

Ha a `lilo.conf` végleges, állítsuk 400-re a *mask*¹⁵⁹-ját, és tegyük rá az *immutable bit*-et.

Az új kernelt teszteljük, stresszeljük először, mielőtt feltennénk az éles szerverre. Persze, mivel a rendszergazdai gép biztosan más hardvereket tartalmaz, mint a szerver, ezért lehet, hogy ki se próbálhatjuk. A kernel fordítása közbeni hibák egy része utalhat hibás memória vagy processzor elemekre. Esetleg nem jó a processzor hűtése. Ellenőrizzük.

Fordítás a kernel-package csomaggal:

Első fázis¹⁶⁰: kernel beszerzése és konfigurálása

¹⁵⁸ Ezt biztonsági okokból mindenképp kötelező betartani. Ha valaki beírja a kernel neve után, hogy „single”, akkor egyfelhasználós üzemmódban indítja el a rendszert és könnyebben szerezhet *root* jogosultságot.

¹⁵⁹ `chmod 400 /etc/lilo.conf`

```
cd <kernel forráskönyvtár>
```

```
make config / make menuconfig / make xconfig, beállítás
```

Második fázis: az /etc/kernel-pkg.conf beállítása

```
maintainer: Kis József      # a csomag készítőjének a neve
email: kjozsi@akarmi.hu    # és a címe
debian: gepnev.datum        # az alapbeállítás, ha véletlenül nem adnánk meg
image_in_boot:true          # ekkor a /vmlinuz szimbolikus link a /boot-ba kerül,
                             # így lehet még egy gyári csomagunk is.
```

Harmadik fázis: Egy szállítható bináris kernel gyártása .deb csomagba

1. `make-kpkg clean` – Ez a parancs letörli az előző próbálkozásaink által generált átmeneti fájlokat. Az `export CONCURRENCY_LEVEL=2` parancssal létrehozunk egy változót, amely a `make` parancsnak megadja a `-j2` paramétert.¹⁶¹
2. (rendszergazdai jogkör), majd `make-kpkg --revision=alfa.20000420 --bzimage kernel_image` – Ez a parancs elkészíti a Debian csomagot. A *revision* paramétere mindig kezdődjön betűvel, hogy egy frissítésnél ne cserélje le a rendszer a saját kernelünket egy „gyárirá”. (Esetleg tegyük „Hold” állapotba.) Ebbe a saját verziószámában nem lehet „_” jel, ne legyen „:” jel, de lehet „-”, „+”, „=” jel. Én a következő módszert javaslom: **gépnév.dátum**. Ekkor egy csomag neve így fog kinézni:

```
kernel-image-2.2.16_2.2.16-alfa.20000420_i386.deb
```

Negyedik fázis: a kernel csomag telepítése. Vigyük át a kernel csomagot a szerverre (pl. `scp`-vel, floppy-val, stb.), majd ott:

```
5. dpkg -i kernel-image-2.2.16_2.2.16-alfa.20000420_i386.deb
```

6. Vizsgáljuk át az `/etc/lilo.conf`-ot, ha kell, tegyük meg a szükséges változtatásokat, majd futtassuk a `lilo`-t.

7. `shutdown -r now` Csak akkor lehet újraindítani a gépet, ha a `lilo` rendesen lefutott.

Végeredményben ez a második változat sokkal kényelmesebb és Debian-kompatibilis. Mindenkinek ezt ajánlom az éles gépeken való használatra. Előnyösebb, mert:

- Kényelmes, hiszen kevesebb parancsot kell kiadni, a script elvégzi helyettünk a munka nagy részét
- Könnyebben tarthatunk több kernelt is egy gépen
- Olyanok készítették el, akik nagyon járatosak a témában, ezért mi nem is hibázhatunk egy kis lépésben sem

¹⁶⁰ `/usr/doc/kernel-package/README.gz` alapján (Manoj Srivastava, svivasta@debian.org)

¹⁶¹ Ezt is tegyük bele a `~/.bashrc`-nkbe.

- A csomagkezelőre bízhatjuk az összes feltelepített programot, nem kell kézi telepítéssel és törléssel vesződni
- Megőrzi a konfigurációs fájlokat későbbi használatra
- Installációs script-ekkel jön, melyek segítik a telepítést és a letörlést is
- Nem kell a modulok másolásával foglalkozni
- Egy gyorsabb gépen is fordíthatjuk a lassabb gép számára a kernelt

Ha kész vagyunk mindennel, és megfelelően le is van tesztelve a kernel, akkor állítsuk a kernel jogait így: `chmod 0400 /boot/kernelneve` Sőt, a jól bevált *immutable bit* is jól jöhet.

2.3 Az Apache finomhangolása, esetleges újrafordítása hardver és feladatorientáltan

Mondhatni ez is egy sokrétű kérdés. A felhasználási – alkalmazási céloktól függ sok minden. Érdemes elolvasni ezt a cikket: [19]. Mivel a munka keretét meghaladja az összes modul funkciójának ismertetése, ezt mellőzöm. Bármilyen felmerülő kérdésünk van, olvassuk el először a dokumentációt (*apache-doc* csomag), vagy nézzük meg a csoport honlapját.

A következőkben a mintapéldához hangolom a rendszert. A példában szereplő cégnek profilja szerint két divíziója van: egy szörme és egy bőrgyártó részleg. Mivel a két divízió eléggé különböző piaccal, PR-el és termékkel rendelkezik ezért két külön Web-helyet hozunk létre nekik. A „fő” Web-hely a cég általános információit tartalmazza és linkeket a két másik, specifikusabb hely felé. Mivel jelen esetben csak egy IP címet vásároltunk, ezért az Apache *NameVirtualHost* funkcióját fogjuk használni. Így képesek leszünk arra, hogy a kliens böngészője által kért Web-helyet adjuk vissza neki név szerint. Mivel az egyetlen IP címünket lefoglalja a *NameVirtualHost*, ezért a fő Web-szerver nem fog szolgáltatni ezen, csak a hurok interfészen. Ezért három virtuális szervert készítünk. Egy a főcég és kettő a két divízió tartalmát tárolja és szolgáltatja.

Az alapbeállítás szerint az Apache-SSL csomag csak a 443-as portot használja. Mi azonban szeretnénk, hogy a szabványos 80-as porton kódolatlan Web-szolgáltatást is nyújtson.

Az Apache konfigurációs állományai az `/etc/apache-ssl/` könyvtárban helyezkednek el. Több állományból áll a rendszer. A fő beállításokat a `httpd.conf` tartalmazza. Az `access.conf` a hozzáférést szabályozza, míg az `srm.conf` állomány a felhasználók által látott rész dolgait szabályozza, mint pl. a MIME, ikonok, stb.

Mivel az Apache esetünkben nagyon fontos, ezért nem csak azokat a beállításokat fogom felsorolni, melyeket meg kell változtatni, hanem nagyrészt azokat is, melyek alapbeállításai jók. Ezt azért teszem, hogy az olvasó minél könnyebben megértse a paraméterek jelentését és szükség szerint finomhangolhassa azt saját igényeihez. A paramétereket kommentezett megjegyzésekkel próbálom érthetővé tenni. Olvassuk el figyelmesen a megjegyzéseket.

Először vegyük sorra a `httpd.conf`-ot:

```
# - Általános beállítások-----
ServerType Standalone      # a szerver indításkor betöltődik, nem az inetd indítja
#Port 443                  #
#Port 80                   #
Listen 443                 # ezeket a portokat figyelje, ez a HTTPS
Listen 80                  # ez a szabványos HTTP port162
HostNameLookups off        # ne keresse meg a kliens nevét, elég az IP cím
                             # a DNS keresés nagyon lelassíthatja a működést

User www-data
Group www-data             # milyen jogokkal fusson a szerver, semmiképp se root!
ServerAdmin webmaster@boresszormegyar.hu # hova küldje a leveleket hiba esetén
ServerRoot /etc/apache-ssl # hol vannak a konfig' fájlok
BindAddress gépünkIPcíme # ezen a címen vagyunk elérhetőek (ha esetleg több IP
                             # címünk is lenne - több hálókártya )
NameVirtualHost gépünkIPcíme:80 # virtuális szervereink ezt az IP címet fogják
NameVirtualHost gépünkIPcíme:443 # figyelni a 80-as és 443-as portokon
# ServerName # mi legyen a fő szerver neve? - erre itt nincs szükség
UseCanonicalName on        # linkek kiegészítése a saját névvel
Timeout 300                # kapcsolat bontása másodpercben
KeepAlive on               # tartós kapcsolatok fenntartása a kliensekkel
MaxKeepAliveRequests 100   # hány db kérést tartson fenn
KeepAliveTimeout 15        # Mennyi másodpercig tartsa fenn a kapcsolatot
MinSpareServers 5          # Minimálisan hány "felesleges" szerver fusson
MaxSpareServers 10         # Maximálisan hány "felesleges" szerver fusson
StartServers 5             # Induláskor hány szervert indítson el
MaxClients 200             # Maximálisan hány kapcsolat élhet
MaxRequestsPerChild 50     # 50 kérés teljesítése után "megöli gyermekét"163
#----- Modulok -----
# Melyik modulokat töltsük be? Azokat, amelyeket én nem láttam szükségesnek
# kikommenteztem. Ha az olvasónak szüksége van valamelyikre, válasszon tetszés
# szerint
# A virtuális szerverek álnév támogatása
LoadModule vhost_alias_module /usr/lib/apache/1.3/mod_vhost_alias.so
# Környezeti változók átadása CGI script-eknek
# LoadModule env_module /usr/lib/apache/1.3/mod_env.so
# Testre szabható naplózás
LoadModule config_log_module /usr/lib/apache/1.3/mod_log_config_ssl.so
# Objektum típus megállapítása tartalomból
```

¹⁶² A `NameVirtualHost` direktíva miatt egyik portot sem figyelhetjük főüzem módban. Ekkor a virtuális szerverek fogják ezeket maguknak fenntartani. A főszerver nem szolgál ki az Internet felé semmit.

¹⁶³ " Egy-egy gyermek által maximálisan kiszolgálható kérések száma. [...] A periodikus leállítással biztosíthatjuk, hogy ha hibásan működik valamelyik gyermek, (vagy az általa indított program) , ne fogyasztassa el az összes memóriát. Linuxon nagyon stabil az Apache, nem fog problémát okozni, ha ez a szám magas." [19. p 90]

```
LoadModule mime_magic_module /usr/lib/apache/1.3/mod_mime_magic.so
# Objektumtípus megállapítása fájlkiterjesztésből
LoadModule mime_module /usr/lib/apache/1.3/mod_mime_ssl.so
# Tartalom „tárgyalás”
LoadModule negotiation_module /usr/lib/apache/1.3/mod_negotiation.so
# a szerver állapotának megjelenítése weblapként (autentikáció szükséges)
LoadModule status_module /usr/lib/apache/1.3/mod_status.so
# Szerver beállítási információk
LoadModule info_module /usr/lib/apache/1.3/mod_info.so
# Szerveroldalon előállított objektumok
LoadModule includes_module /usr/lib/apache/1.3/mod_include.so
# Automatikus könyvtár listázás
LoadModule autoindex_module /usr/lib/apache/1.3/mod_autoindex.so
# Alapszintű könyvtárkezelés
LoadModule dir_module /usr/lib/apache/1.3/mod_dir.so
# CGI script-ek meghívása
LoadModule cgi_module /usr/lib/apache/1.3/mod_cgi.so
# Az „.asis” fájl kezelő
LoadModule asis_module /usr/lib/apache/1.3/mod_asis.so
# Kép-térkép fájlkezelő
LoadModule imap_module /usr/lib/apache/1.3/mod_imap.so
# Automatikus hibajavítás félregévelt url-ekben
LoadModule speling_module /usr/lib/apache/1.3/mod_speling.so
# A felhasználók könyvtárait kezeli (listázás, letöltés)
LoadModule userdir_module /usr/lib/apache/1.3/mod_userdir.so
# HTTP gyorsítótár
LoadModule proxy_module /usr/lib/apache/1.3/libproxy.so
# Álnevek és átirányítások
LoadModule alias_module /usr/lib/apache/1.3/mod_alias.so
# Az access.conf értelmezéséhez
LoadModule access_module /usr/lib/apache/1.3/mod_access.so
# Felhasználó azonosítás szöveg fájlokkal
LoadModule auth_module /usr/lib/apache/1.3/mod_auth_ssl.so
# ftp stílusú anonimusz felhasználó azonosítás
LoadModule anon_auth_module /usr/lib/apache/1.3/mod_auth_anon.so
# felhasználó azonosítás DBM fájlokkal
LoadModule dbm_auth_module /usr/lib/apache/1.3/mod_auth_dbm.so
# felhasználó azonosítás a Berkeley-féle DB (adatbázis) fájlokkal
LoadModule db_auth_module /usr/lib/apache/1.3/mod_auth_db.so
# MD5 felhasználó azonosítás
LoadModule digest_module /usr/lib/apache/1.3/mod_digest.so
# HTTP fejléc metafájlok támogatása
LoadModule cern_meta_module /usr/lib/apache/1.3/mod_cern_meta.so
# Fejlécek erőforrásokhoz (pl. meddig érvényes egy oldal)
LoadModule expires_module /usr/lib/apache/1.3/mod_expires.so
# Tetszőleges HTTP fejlécek beépítése
LoadModule headers_module /usr/lib/apache/1.3/mod_headers.so
# Felhasználó-követés sütik segítségével
LoadModule usertrack_module /usr/lib/apache/1.3/mod_usertrack.so
# egységes kérés azonosító generálása minden lekéréshez
LoadModule unique_id_module /usr/lib/apache/1.3/mod_unique_id.so
# Környezeti változók beállítása kliens információk alapján
LoadModule setenvif_module /usr/lib/apache/1.3/mod_setenvif.so
# Szerverek online statisztikái.
LoadModule throttle_module /usr/lib/apache/1.3/mod_throttle.so
```



```
# SSL rendszerhívások
AddModule apache_ssl.c
# LDAP alapú azonosítás
# LoadModule auth_ldap_module /usr/lib/apache/1.3/auth_ldap.so
# Eszlőzfájlok kezelése
# LoadModule allowdev_module /usr/lib/apache/1.3/mod_allowdev.so
# PostgreSQL adatbázis alapú azonosítás
# LoadModule pgsql_auth_module /usr/lib/apache/1.3/mod_auth_pgsql.so
# A fontos PHP3 modul
LoadModule php3_module /usr/lib/apache/1.3/libphp3.so
# Netscape 4.x roaming támogatása
# LoadModule roaming_module /usr/lib/apache/1.3/mod_roaming.so

#----- Naplózás -----
# itt tárolja induláskor a folyamat azonosító számát
PidFile /var/run/apache-ssl.pid
ErrorLog /var/log/apache-ssl/error.log # A hibaüzenetek ide menjenek
LogLevel warn # mit loggoljon? debug/info/notice/warn/error/crit
# mi legyen a naplózás formája?
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\" %T %v" full
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\" combined
LogFormat "%h %l %u %t \"%r\" %>s %b" common
LogFormat "%{Referer}i -> %U" referer
LogFormat "%{User-agent}i" agent

# Egyéni naplófájlok:
# Vigyázat! Ha nagy a forgalom nagyra dagadhat a fájl. Biztonsági szempontból
# előnyös, hiszen minden egyes kérés naplózva lesz IP címmel. Ha támadás gyanú van,
# akkor mindenképp kapcsoljuk be. Továbbá a statisztikákat is ezekből készítjük.
CustomLog /var/log/apache-ssl/access.log common # naplózhatjuk a hozzáférést is
# Ezzel naplózzuk azt az oldalt, ahonnan idelépett a kliens
# CustomLog /var/log/apache-ssl/referer.log referer
# A kliens böngészőinformációit naplózza
# CustomLog /var/log/apache-ssl/agent.log agent

# SSL beállítások -----
#Globális szerver-gyorsító a kulcsok számára
SSLCacheServerPath /usr/lib/apache-ssl/gcache
SSLCacheServerPort /var/run/gcache_port # a használandó port (unix domain socket)
SSLSessionCacheTimeout 50 # időzítő
SSLCACertificatePath /etc/apache-ssl # a CA igazolás helye
#SSLCACertificateFile CA.pem # ha van saját CA igazolásunk
SSLCertificateFile /etc/apache-ssl/apache.pem # az igazolás

# SSLVerifyClient beállítása:
# 0 ha nincs szükség arra, hogy a kliensnek legyen igazolása
# 1 ha a kliens adhat igazolást
# 2 ha a kliensnek kötelezően be kell mutatnia egy igazolást
# 3 ha a kliensnek lehet igazolása, de nem szükséges, hogy az érvényes CA-ja legyen
SSLVerifyClient 0
SSLFakeBasicAuth # a BD és DBM kompatibilitás miatt
#SSLRequireCipher # csak ilyen titkosító algoritmusokat fogadunk el
#SSLBanCipher # ezeket pedig letiltjuk
# SSL tranzakciók naplózása:
CustomLog /var/log/apache-ssl/ssl.log "%t %{version}c %{cipher}c %{clientcert}c"
```

```
# ---Virtuális gépek -----
# Itt definiáljuk a két gyárrészleg külön oldalait
<VirtualHost www.szormegyar.hu:80> # virtuális szerver ezen a néven a 80-as porton
#ServerAdmin                      # lehetne külön webmestere mindnek
SSLDisable                        # itt kikapcsoljuk az SSL-t
DocumentRoot /var/www/szormegyar # mindenkinek a saját gyökerét
ServerName www.szormegyar.hu     # és a saját nevét adjuk meg
ServerAlias szormegyar.hu *.szormegyar.hu # az erre hivatkozókat is ide irányítjuk
# külön is kérhetjük a log-okat.
ErrorLog /var/log/apache-ssl /szormegyar-error.log
# Az összes kérést és forgalmat is lehet külön naplózni,
# ebből készülnek a statisztikák.
CustomLog /var/log/apache-ssl/szormegyar-access.log common
#TransferLog /var/log/apache-ssl/szormegyar-access.log
</VirtualHost>

<VirtualHost www.szormegyar.hu:443> # ugyanazt kikínáljuk SSL-el is
SSLEnable
DocumentRoot /var/www/szormegyar
ServerName www.szormegyar.hu
ServerAlias szormegyar.hu *.szormegyar.hu
ErrorLog /var/log/apache-ssl/szormegyar-ssl-error.log
CustomLog /var/log/apache-ssl/szormegyar-ssl-access.log common
</VirtualHost>

<VirtualHost www.borgyar.hu:80>
DocumentRoot /var/www/borgyar
SSLDisable
ServerName www.borgyar.hu
ServerAlias borgyar.hu *.borgyar.hu
ErrorLog /var/log/apache-ssl/borgyar-error.log
CustomLog /var/log/apache-ssl/borgyar-access.log common
</VirtualHost>

<VirtualHost www.borgyar.hu:443>
SSLEnable
DocumentRoot /var/www/borgyar
ServerName www.borgyar.hu
ServerAlias borgyar.hu *.borgyar.hu
ErrorLog /var/log/apache-ssl/borgyar-ssl-error.log
CustomLog /var/log/apache-ssl/borgyar-ssl-access.log common
</VirtualHost>

<VirtualHost www.boresszormegyar.hu:80>
DocumentRoot /var/www/boresszormegyar
SSLDisable
ServerName www.boresszormegyar.hu
ServerAlias boresszormegyar.hu *.boresszormegyar.hu
ErrorLog /var/log/apache-ssl/boresszormegyar-error.log
CustomLog /var/log/apache-ssl/boresszormegyar-access.log common
</VirtualHost>

<VirtualHost www.boresszormegyar.hu:443>
SSLEnable
```

```

DocumentRoot /var/www/boresszormegyar
ServerName www.boresszormegyar.hu
ServerAlias boresszormegyar.hu *.boresszormegyar.hu
ErrorLog /var/log/apache-ssl/boresszormegyar-ssl-error.log
CustomLog /var/log/apache-ssl/boresszormegyar-ssl-access.log common
</VirtualHost>

```

A következő fájl, az `access.conf`:

Minden könyvtárra külön szabályokat állíthatunk fel.

```

<Directory /var/www>
# none: semmi, all: az összes következő, Indexes: a kliens listázhatja,
# Includes: futtathat szerver-oldali scripteket,
# FollowSymLinks: követi a szimbolikus linkeket - veszélyes lehet!
# ExecCGI: CGI scripteket futtathatunk innen - veszélyes!
# olvassuk el a dokumentációt! /usr/share/doc/apache/manual/mod/core.html#options
Options none
AllowOverride none # olvassuk el a dokumentációt!
Order deny,allow
# deny from gonosz.betörő.domén
allow from all
</Directory>

#A CGI-ket tartalmazó könyvtár. Nézzük meg mi van benne. Ha üres, akkor
# kommentezzük ki innen is.
<Directory /usr/lib/cgi-bin>
AllowOverride None
Options ExecCGI FollowSymLinks
</Directory>

```

Az alkönyvtárak ezeket a beállításokat öröklik. Az alkönyvtárak beállításait egyenként felülbírálnak.

Ha egyes könyvtárakhoz való hozzáféréshez felhasználó azonosítást szeretnénk, megtehetjük az `access.conf` fájlban, vagy az adott könyvtárban elhelyezett `.htaccess/.htpasswd` fájlban is. A kódoláshoz és azonosításhoz többféle autentikációs modul használható:

<i>mod_auth</i>	alapfunkciós, <i>crypt()</i> függvényt használ
<i>mod_digest</i>	MD5 kódolást használ
<i>mod_auth_sys</i>	a <code>/etc/passwd</code> fájlt használja
<i>mod_auth_pam</i>	a PAM függvényeit használja

10. táblázat - Autentikációs modulok

Nekünk a PAM lenne jobb, de az `/etc/shadow` fájlt nem látja, ezért azt olvashatóvá kellene tenni az Apache számára, ami komoly veszélyeket rejt magában. Az árnyékjelszó miatt az *auth_sys* sem működik sajnos. Az MD5-ös kódolás jó választás lenne, de a legtöbb böngésző (pl. Netscape 4.x) még nem támogatja. Így kénytelenek vagyunk a hagyományos azonosítást használni. Itt használhatunk sima szöveges, DB

és DBM formátumú fájlokat is. Mivel nem sok felhasználót kell tárolnunk, megfelel a szöveges fájl is. Egy jó cikk olvasható kezdőknek e témában itt [20].

Ha betöltöttük a *status* modult, akkor Weben keresztül is lekérdezhető a szerver állapota. Vigyázat, ennek nyilvánossá tétele megkönnyíti a betörők helyzetét! Alkalmazzunk SSL-es kódolást és azonosítást. Csak a rendszer és Web-gazdák tekinthessék meg a szerver állapotát. Ehhez először is létre kell hoznunk egy jelszófájlt. Ez a fájl nem lehet a */var/www* könyvtár alatt, hiszen ekkor azt mások könnyen meg tudják szerezni. Helyezzük el az */etc/apache-ssl* könyvtárban, pl. *info.pwf* néven¹⁶⁴. Így hozhatjuk létre:

```
htpasswd -c jelszófájl felhasználó; chmod 0640 jelszófájl;  
chgrp www-data jelszófájl
```

Vigyázzunk, hogy ha más autentikációs modul is be van töltve, az lesz érvényes és ezért esetleg nem fog helyesen működni a *mod_auth*.

```
<Location /szerver-status>  
SetHandler server-status      # ez esetben a server-status modul veszi át a tartalmat  
order deny,allow  
#deny from all  
allow from all # ide azt írjuk be, hogy honnan lehessen lekérdezni.  
SSLRequireSSL                # ide csak SSL-el lehet belépni  
# AuthPam_Enable off         # ha betöltöttük a mod_auth_pam modult, most kapcs. ki  
AuthType Basic                # crypt() függvényt használó azonosítás  
AuthName Státusz              # ez jelenik meg a jelszó kérésekor  
AuthGroupFile /dev/null       # a csoport lista most üres (nem kötelező)  
AuthUserFile /etc/apache-ssl/info.pwf # ebben a fájlban tároljuk a jelszavakat  
require user rgazda wgazda   # csak ezek a felhasználók férhetnek hozzá az infókhoz  
</Location>  
  
# A szerver beállításait tartalmazó információkkal is ugyanígy járjunk el:  
<Location /szerver-info>  
SetHandler server-info  
order deny,allow  
#deny from all  
allow from all # ide azt írjuk be, hogy honnan lehessen lekérdezni.  
SSLRequireSSL                # ide csak SSL-el lehet belépni  
# AuthPam_Enable off         # ha betöltöttük a mod_auth_pam modult, most kapcs. ki  
AuthType Basic  
AuthName Infó  
AuthUserFile /etc/apache-ssl/info.pwf  
require user rgazda wgazda  
</Location>  
  
# Ez a phf típusú betörési kísérleteket naplózza  
Location /cgi-bin/phf*>  
deny from all  
ErrorDocument 403 http://phf.apache.org/phf_abuse_log.cgi  
</Location>
```

¹⁶⁴ Csak a root tudja írni, és a *www-data* csoport olvasni!

```
# Kiadhatjuk a dokumentációt, hogy könnyebb legyen a konfigurálás.
<Directory /usr/share/doc>
Options Indexes FollowSymLinks
AllowOverride None
order allow,deny
allow from 127.0.0.1 "rendszergazdaIP-je"
deny from all
SSLRequireSSL          # ide csak SSL-el lehet belépni
AuthType Basic
AuthName Doksik
AuthUserFile /etc/apache-ssl/info.pwf
require user rgazda wgazda
</Directory>
Alias /doksik/ /usr/doc/
# Ez is veszélyes lehet, mert a betörő megtudhatja, hogy milyen programok
# vannak a gépen, ezért mindenképp kössük azonosításhoz a hozzáférést.

<location /th-info> # az egyes virtuális gépek forgalmát monitorozhatjuk vele.
SetHandler throttle-info
order deny,allow
#deny from all
allow from all # ide azt írjuk be, hogy honnan lehessen lekérdezni.
SSLRequireSSL          # ide csak SSL-el lehet belépni
# AuthPam_Enable off    # ha betöltöttük a mod_auth_pam modult, most kapcs. ki
AuthType Basic
AuthName Th-Infó
AuthUserFile /etc/apache-ssl/info.pwf
require user rgazda wgazda
</location>

#A felhasználók - bár nem lesznek - ne tudjanak symlink-támadást csinálni.
<DirectoryMatch ^/home/.*/public_html>
Options Indexes SymLinksIfOwnerMatch
AllowOverride None
</DirectoryMatch>

# Biztonsági okokból ezeknek a fájloknak a forgalmát letiltjuk
<Files .htaccess>
order allow,deny
deny from all
</Files>
<Files .htpasswd>
order allow,deny
deny from all
</Files>
```

Az `srm.conf`-ot itt nem részletezem, mert alapbeállításai jók számunkra, mindazonáltal mindenki olvasgassa át, és ha szüksége van rá, ízlés szerint módosítsa.

Az Apache fordítása

Ezt csak és kizárólag olyanok végezzék, akiknek már van kellő tapasztalatuk és gyakorlatuk az Apache fordításában, annak konfigurálásában, általánosan a fordítások folyamatában, és tényleg tudják, hogy mit csinálnak. Sajnos ez elég nehéz kezdetben. Amennyiben úgy döntünk, hogy lefordítjuk magunknak az Apache-ot, mert:

- Nagy forgalmat fogunk bonyolítani és a processzorhoz akarjuk optimalizálni a kódot, a gyorsabb végrehajtás érdekében
- Olyan modulra van szükségünk, amelyet a “gyári” csomag nem tartalmaz
- Az 1.3.12-es vagy újabb verzióra van szükségünk az új funkciók miatt
- Egyszerűen szeretünk mindent a saját kezünkben tartani

akkor üljünk át a rendszergazdai géphez.

- Ha megfelel az 1.3.9-es verzió és az `apt`-t úgy konfiguráltuk, hogy a forráskód csomaglistát is kezelje, akkor elég ezt a parancsot kiadnunk: `apt-get source apache-ssl` Ha minden jól ment, akkor néhány percen belül az `/usr/src` alatt találjuk a szükséges fájlokat.¹⁶⁵
- Ha az 1.3.12-es vagy újabb változatra van szükségünk, akkor adjuk ki a következő parancsokat (vagy töltsük le `ftp`-vel a következő fájlokat):

```
wget ftp://ftp.hu.debian.org/debian-non-US/dists/woody/non-US/main/source/apache-ssl*
és helyezzük el az /usr/src könyvtárba.
```

Ezek után lépünk be az `/usr/src` könyvtárba és adjuk ki ezt a parancsot: `dpkg-source apache-ssl*.dsc`¹⁶⁶ Ez kibontja a három forráscsomagot. Lépünk be az `apache-ssl*` alkönyvtárba. Ne felejtjük el ellenőrizni a `DEBIAN_BUILDARCH` változó értékét.¹⁶⁷ Ha a fordítás előtti konfigurációt is módosítani szeretnénk, akkor először is tudnunk kell, hogy mit is akarunk pontosan. Pl. ha az összes modult le szeretnénk fordíttatni, nem csak a stabil állapotúakat, lépünk be a `debian` alkönyvtárba és szerkesszük meg a `rules` fájlt a következőképp: írjuk be a 100. sorába az *all* szót a *most* helyébe:

¹⁶⁵ Ha nem állítottuk be rendesen az `apt`-t, akkor írjuk be a következőket a rendszergazda gépén az `/etc/apt/sources.list` fájlba:

```
deb-src ftp://ftp.hu.debian.org/debian potato main contrib non-free
```

```
deb-src ftp://ftp.hu.debian.org/debian-non-US potato non-US/main non-US/contrib non-US/non-free
```

Majd futtassuk az `apt-get upgrade` parancsot.

¹⁶⁶ Vagy egészítsük ki a megfelelő verziószámokkal.

¹⁶⁷ `set | grep DEBIAN`

17. kép - rules fájl szerkesztése

Ha nem akarjuk megváltoztatni a modulok konfigurációját, akkor lépünk tovább. Ezután lépünk vissza egy könyvtárral, majd: `debian/rules binary` paranccsal indíthatjuk a fordítást. Ha minden fordításhoz szükséges csomag fenn van a gépen és van elég helyünk is, akkor pár perc múlva (lassabb gépen tovább is eltarthat!) az `/usr/src` könyvtárban megjelennek a kész, lefordított csomagok. Ezután a `dpkg -i csomagnév` paranccsal telepíthetjük őket. Ha már kész, beállított konfigurációs állományaink vannak, ne engedjük felülírni őket! Ha minden jól ment, akkor már egy gépre optimalizált, egyéni beállításokkal fordított Apache szerverünk fut a gépen.

2.4 Az SSH konfigurációjának finomhangolása

A Debian-ban az OpenSSH program alapbeállításai elég jók és szigorúak, ám ezen is lehet még finomítani. Az `/etc/ssh` könyvtárban lévő `ssh_config` és `sshd_config` fájlokat kell megvizsgálnunk. Az első a kliens a második a szerver program beállításait tartalmazza. Érdeemes először a manuál oldalakat elolvasni, mert így legalább tisztában leszünk a következő fogalmakkal. Az `ssh` képes X11 kapcsolatokat és más TCP portokat továbbítani a titkosított csatornán.

- Ha léteznek a `/etc/hosts.equiv` vagy `/etc/ssh/shost.equiv` fájlok (megfelelően ki is vannak töltve) és mindkét gépen a felhasználónak ugyanaz a *login* neve, akkor azonosítás nélkül beléphet az egyik gépről a másikra. Ezt nekünk mindenképp le kell tiltanunk, mert komoly biztonsági veszélyforrást jelent. (*RhostsAuthentication*)
- Ez az eljárás kiegészülhet azzal, hogy a szerver ellenőrizheti a kliens gépének RSA kulcsát és csak ezek után engedélyezi a belépést. Ez már valamennyire biztonságosabb az előzőnél, de még mindig fennáll annak a veszélye, hogy valaki más ül a kliens előtt, mint akié a számla és ekkor az a

jelszó ismerete nélkül is beléphet. Ezért ez a módszer is kerülendő. (*RhostsRSAAuthentication*)

- A harmadik azonosítási módszer a tiszta RSA autentikáció, mely nyilvános kulcsú azonosítást tesz lehetővé. Ez is felment minket a jelszók állandó gépelésétől, viszont generálni kell minden egyes felhasználónak (mindenki saját magának) egy saját titkos és nyilvános kulcsot. Ezután, ha a szerveren megvan a megfelelő kulcs párja, akkor egy kulcsteszt után beenged. (*RSAAAuthentication*)
- A negyedik és egyben legbiztonságosabb azonosítási rendszer a hagyományos jelszó alapú rendszer. (*PasswordAuthentication*)

Figyelem! Minden felhasználó felülbíráhatja a gépszintű beállításokat a saját `.ssh/config` fájl beállításával.

Nézzük tehát a kliens beállításait: `/etc/ssh/ssh_config`

```
Host localhost                # ezek a saját gépre vonatkoznak.
    ForwardAgent yes          # csak itt engedélyezzük az X11-es titkosított
    ForwardX11 yes            # csomagok átjátszását. Ha ezt engedélyeznénk
Host *                        # kívülre is, akkor komoly biztonsági problé-
    ForwardAgent no           # val kellene szembenézni.
    ForwardX11 no
    RhostsAuthentication no   # .rhost fájl alapján ne engedjünk azonosítást
    RhostsRSAAuthentication no # még RSA kulcsosat se
    RSAAAuthentication yes    # RSA kulcsos azonosítás jelszó helyett
    PasswordAuthentication yes # a jelszós azonosítást engedélyezzük
    FallBackToRsh no          # hiba esetén ne térjünk vissza kódolatlan üzemmódba
    UseRsh no                  # rsh használatát tiltsuk le
    BatchMode no               # ez a script-ekben használatos, de tiltsuk le, mert
                                # ekkor nincs jelszó-kérés!
    CheckHostIP yes           # ha név szerint kérjük, ellenőrizzük le a név-IP párost
# StrictHostKeyChecking yes   # nem engedjük új gépek kulcsának elfogadását
    IdentityFile ~/.ssh/identity # itt van a saját kulcsunk
    Port 22                    # általában itt kell keresgelnünk
    Cipher blowfish            # ez a titkosító algoritmus sokkal gyorsabb és bizton-
                                # ságosabb, mint a DES vagy a 3DES.
    EscapeChar ~               # bővebb leírást ezekről a manuálban!
    Compression yes            # ha modemén keresztül vagyunk, akkor jól jöhet
                                # a tömörítés, gyors hálózaton viszont lassít.
    GatewayPorts no            # más gépek ne kapcsolódhassanak a helyi továbbított
                                # portokhoz
```

Ez itt pedig egy minta `sshd_config`:

```
Port 22                        #melyik portot használjuk
ListenAddress 0.0.0.0         # milyen tartományból engedjünk be hívásokat
HostKey /etc/ssh/ssh_host_key # a szerver kulcsa
ServerKeyBits 1024 #eredeti: 768 # a kulcs hossza
LoginGraceTime 300            # hány másodperce van a kliensnek azonosításra
KeyRegenerationInterval 3600 # a kulcs óránként újragenerálódik
PermitRootLogin no            # a root nem jelentkezhet be!
#AllowGroups webcsap          # csak ezek a csoportok jelentkezhetnek be
```



```
#AllowUsers webgazda system      # csak ezek a felh. jelentkezhetnek be
#DenyGroups stuff nogroup        # ezek semmiképp sem jelentkezhetnek be
#DenyUsers nobody
StrictModes yes                  # ellenőrzi belépés előtt a felh. fájlok jogait
X11Forwarding yes                # szerverszinten engedélyezhető
X11DisplayOffset 10
KeepAlive yes                    # próbálja meg életben tartani a kapcsolatot
UseLogin no                      # ne használjuk, hátha trojan-os
PrintMotd no                     # mivel ezeket már a PAM kezeli, kikapcs.
CheckMail no
SyslogFacility AUTH              # melyik syslog kategóriába tartozzon az
                                  # azonosítás
LogLevel INFO                    # naplózási szint
RhostsRSAAuthentication no
IgnoreRhosts yes                 # el se olvassuk ezeket
RhostsAuthentication no
IgnoreUserKnownHosts yes         # ekkor csak a rendszergazda határozhatja meg,
                                  # melyek az ismert gépek
RSAAuthentication yes
PasswordAuthentication yes
PermitEmptyPasswords no          # ne engedjünk meg üres jelszavakat!
```

Ezek után indítsuk újra az sshd-t az `/etc/init.d/ssh restart` paranccsal. Teszteljük le az új beállításokat és próbáljunk bejelentkezni másik gépekről a szerverre. Saját RSA kulcsot az `ssh-keygen` paranccsal hozhatunk létre. Mindenképp válasszunk egy jó jelszót ehhez a kulcshoz is.

2.6 Szoftveres figyelő („watchdog”) beállítása

Ha a rendszerünket pl. DoS támadás éri, akkor nagyon nagy terhelésnek lesz kitéve. A CPU terhelési mutatója 25 fölé is mehet. Ez esetben jobb, ha a rendszer újraindítja magát, ezzel újra életképesse téve önmagát – és persze a DoS támadást is hatástalanítja egy időre ilyenkor. A rendszer terhelésének monitorozására sokféle hardverkártyás megoldás is van, melyek sokkal hatékonyabbak, mint az ingyenes szoftveres megoldás: ha a rendszer teljesen lefagy, akkor is újra indítja a hardverkártya. Ha belefér a keretbe, vásároljunk egy Linux-kompatibilis (lásd: kernel dokumentáció) hardverkártyát. Ha nincs rá lehetőségünk a kernel lehetőséget ad szoftveres terhelésfigyelésre is. A `watchdog` csomag tartalmaz egy azonos nevű démont, mely elég jól testre szabható és működik a kernel által támogatott összes hardveres és szoftveres megoldással is. A `man watchdog` és a `man watchdog.conf` parancsok segítségével olvassuk el a dokumentációt.

A program segítségével rengeteg mindent lehet monitorozni, pl.:

- a rendszer terhelése az elmúlt 1, 5, 15 percben nem ment-e adott értékek fölé
- egy általunk beállított memória mennyiség szabad-e
- a rendszer hőmérséklete elérte-e a határszintet (csak hardverkártyák esetén)
- egy adott fájlba tud-e írni, vagy azon történt-e változás (pl. naplófájlok)

- fut-e az adott szerverprogram (*PID*-fájlon keresztül)
- él-e a hálózaton valamely gép vagy tartomány (*ping*)
- egy adott hálózati interfészen van-e forgalom (*eth*)

A program segítségével saját teszt-program és helyreállító / leállító program futtatható. Ha leállítás vagy újraindítás történt, akkor e-mail-ben értesíti az adott személyt. Szerkesszük meg tehát az `/etc/watchdog.conf` fájlt:

```
admin rgazda@gyakranolvasom.hu
max-load-1 = 24          # 24-es terhelésnél indítson újra
max-load-5 = 18
max-load-15 = 12
watchdog-device=/dev/watchdog      # ez a kernel eszköze
```

Igény szerint adjuk meg saját opcióinkat is. Indítsuk újra a démon-
`/etc/init.d/watchdog restart`

Figyelem! Ha esetleg 2.3.x vagy 2.4.x sorozatú kernelt használunk, ne lepődjünk meg, ha váratlanul, minden ok nélkül újraindul a rendszer, mert a `watchdog` programot még nem hangolták az új kernelekhez. Ez esetben inkább távolítsuk el a programot.

2.7 E-mail titkosító kulcspárok létrehozása a „gpg” programmal

A következő fontos feladat: saját kulcspár létrehozása mindkét gépen a `gpg` (*GNU Privacy Guard*) segítségével. Ez azért fontos, mert ha e-mail-jeinket lehallgatják, sok mindent megtudhatnak a szerverről. Pl. ha a naplófájlokat kódolatlanul továbbítjuk, bárki elkaphatja és elolvashatja azokat. A nyilvános kulcsú titkosítási eljárást nem részletezem, túlmutat a témán. Nézzünk utána, pl. a [12. p. 185.], vagy a GPG dokumentációjában.

Ha még nem telepítettük, telepítsük mindkét gépre a `gnupg` csomagot. A `pgp`-vel való kompatibilitás miatt van néhány kiegészítő csomagja is `gpg-*` néven. Ha szükség van rá, telepítsük ezeket is. Olvassuk el a dokumentációt.

Első lépésként a rendszergazda gépén készítsünk egy kulcspárt: Az alábbi paranccsal indíthatjuk az interaktív folyamatot:

```
[rgazda@rgazdagepe:~]#gpg --gen-key
gpg (GnuPG) 1.0.1; Copyright (C) 1999 Free Software Foundation, Inc.
This program comes with ABSOLUTELY NO WARRANTY.
This is free software, and you are welcome to redistribute it
under certain conditions. See the file COPYING for details.
```

Válasszuk ki, milyen kódolási algoritmusokat használjunk (1)

```
Please select what kind of key you want:
(1) DSA and ElGamal (default)
(2) DSA (sign only)
(4) ElGamal (sign and encrypt)
```

```
Your selection? 1
DSA keypair will have 1024 bits.
About to generate a new ELG-E keypair.
```

Válasszunk legalább 2048 bites kulcsot, de a 4096 bit az ajánlott méret, mivel ezt már tényleg nehéz feltörni (gondoljunk a jövő gépeire is!).

```
        minimum keysize is 768 bits
        default keysize is 1024 bits
    highest suggested keysize is 2048 bits
What keysize do you want? (1024) 2048
Do you really need such a large keysize? y
Requested keysize is 2048 bits
```

A kulcs érvényessége ne járjon le. Bár javasolt legalább évente újat létrehozni.

```
Please specify how long the key should be valid.
    0 = key does not expire
    <n> = key expires in n days
    <n>w = key expires in n weeks
    <n>m = key expires in n months
    <n>y = key expires in n years
Key is valid for? (0)
Key does not expire at all
Is this correct (y/n)? y
```

Itt megadjuk a felhasználó nevét és e-mail címét.

```
You need a User-ID to identify your key; the software constructs the user id
from Real Name, Comment and Email Address in this form:
    "Heinrich Heine (Der Dichter) <heinrichh@duesseldorf.de>"

Real name: Kis Jozsef
Email address: rgazda@gyakranolvasom.hu
Comment:
You selected this USER-ID:
    "Kis Jozsef <rgazda@gyakranolvasom.hu>"

Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? o
```

Adjunk meg kétszer egy jó és biztonságos jelszót, majd jegyezzük fel valahova, ahol megtaláljuk, mert úgyis el fogjuk felejteni.

```
You need a Passphrase to protect your secret key.

Enter passphrase:
```

Ezután a gép elkezd legenerálni a kulcsokat, ez elég sokáig is eltarthat, ha lassú a gépünk. Jobb, ha minél gyorsabb gépen végezzük el ezt. Arra kér, hogy mozgassuk az egeret, vagy tegyünk valamit, hogy minél jobb entrópiát hozhasson létre. Jelentkezzünk át egy másik terminálra és csináljunk valami hasznosat.

```
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
```

```
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
.....
+++++
```

Amint kész a folyamat, exportáljuk ki a kulcsot egy szövegfájlba:

```
gpg --export -a -u Kis -o kisjozsi.gpg.txt
```

Ezután a szerveren is generáljunk egy kulcspárt a *root* számára (root@alfa.boresszormegyar.hu). Ha ez megtörtént, akkor a szerverre vegyük fel az előbbi kiexportált nyilvános kulcsot.

```
[root@alfa:~/]cat kisjozsi.gpg.txt | gpg --import
```

Ekkor, ha minden jól ment, az olvasós e-mail címünk kulcsa felkerült a kulcscsomóra. Ennek a hasznára majd később visszatérek.

2.8 A rendszernapló (log) és kezelése

A rendszergazda számára a rendszerről az elsődleges információforrás a rendszernapló (*log*). Ebből tudhatja meg, hogy mi történt a rendszerén, amíg távol volt. Nagyon fontos ezt a naplót karbantartani, kezelni, olvasgatni. Ebből derülhetnek ki a betörési kísérletek is. Ha a betörő nagyon ügyes, akkor meg tudja változtatni ezeket a fájlokat. Ekkor észrevétlen maradhat a számunkra. Ezért fontos, hogy automatizáljuk a naplózási rendszerünket. Ha tehetjük (van a hálózatban egy log-szerver) akkor tükrözzük az információkat. A másik lehetőség egy régi leprellős mátrix nyomtató alkalmazása, mint másodlagos log-kimenet. Ekkor minden információ papírra kerül és ez bizonyíték értékű lehet betörés esetén. (Hátránya persze a kifogyó papír.)

Először is állítsuk be úgy a `/var/log` könyvtár jogait, hogy az egyéb felhasználók *ne is léphessenek oda be*.¹⁶⁸ Továbbá azok a programok, melyek nem a `syslog()`-on keresztül naplóznak, legyen bejárásuk ide, különben el sem indulnak.

```
[root@alfa:/var]#chmod o-rwx log; addgrp loggers; chgrp loggers log; adduser mysql
loggers; adduser www-data loggers;
[root@alfa:/var]#ls -ld log
drwxr-x--- 12 root loggers 8192 APR 28 09:39 log
```

Ez azért fontos, hogy egy felhasználó jelszavát megszerezve ne tudjon a behatoló információkat szerezni a rendszerről. Ekkor viszont vigyáznunk kell, mert azok a programok, melyek nem a `syslog()`-on keresztül naplóznak és nem *root*-ként futnak, nem tudnak majd hozzáférni a napló-fájljaikhoz, kivéve, ha hozzá nem adjuk őket a *loggers* csoporthoz.

¹⁶⁸ Vigyázat! Ekkor csak a *root* joggal futó démonoknak lesz joguk ide (a `syslog()`-on keresztül) naplózni. Ezt úgy védhetjük ki, hogy létrehozunk egy *loggers* nevű csoportot (`addgroup loggers`), melynek `chgrp loggers /var/log` paranccsal átadhatjuk a könyvtárat. Ezután minden olyan démont, amely nem a `syslog` segítségével naplóz és saját felhasználói azonosítóval rendelkezik, adjunk hozzá ehhez a csoporthoz az `adduser név loggers` paranccsal.

2.8.1 A „syslog” démon kiválasztása

A Potato-ban két *syslog* démon közül választhatunk. Az első a hagyományos *syslogd*. Ezt eredetileg a BSD-ből hozták át. Ez a rendszer található szinte minden UN*X típusú rendszeren. A másik lehetőség az új, magyar fejlesztésű *syslog-ng* (*New Generation*). Ez sokkal korszerűbb, a mai igényeknek megfelelően kialakított, már stabil állapotú, könnyen konfigurálható program. Egy fiatal magyar programozó fejlesztte, *Scheidler Balázs*, GPL licenz alatt. Ez szinte minden UN*X típusú rendszer alatt használható. Az én javaslatom, hogy ezt válasszuk előnyei miatt. (Bővebb információként nézzük meg a dokumentációt.) A továbbiakban erre fogok hivatkozni. A Debian-ban a *syslog-ng* alapból megfelelően be van konfigurálva, hacsak nem akarunk log-szervert használni. Egyedi kívánságokért nézzük át a konfigurációs állományokat.

2.8.2 A naplófájlok rotálásának beállítása

A rendszernapló a használat során egyre csak duzzad, s ha nem töröljük le időnként, akkor teljesen be fogja tölteni a */var* partíciót. Ennek a kiküszöbölésére írták a *logrotate* programot. Megfelelő beállítás után időközönként átnevezi a naplófájlokat, megszámozza őket és megtart annyit, amennyit meghatároztunk neki, a többit pedig letörli. Állítsuk be az */etc/logrotate.conf*-ot a következőképp:

```
# először olvassuk el a logrotate manuál oldalát!
# az itt található beállítások globálisak, majd
# külön-külön minden fájlra definiálhatunk szabályokat
# ----- Globális
# a naplófájlokat naponta rotáljuk
daily
# 14 napot megtartunk, azután töröljük. (minél több megmarad, annál jobb a biztonság
# szempontjából, de annál rosszabb a hely szempontjából.)
rotate 14
# a hibákat ide küldje
errors rgazga@gyakranolvasom.hu
# új üres naplófájlok létrehozása rotálás után, megadható, hogy milyen jogokkal
# keletkezzen az új és ki legyen a tulajdonosa: create mód tulaj csoport
create 027 # ekkor a „többiek” számára nem lesz olvasható
# a rotált log-okat letömörítjük, hogy kevesebb helyet foglaljanak
compress
# ha üres a fájl, akkor ne forgassuk
noifempty
# egyes csomagok ebbe az alkönyvtárba teszik a logrotate-ra tartozó információkat
# ezért azokat így érvényesítjük.
include /etc/logrotate.d
# ide küldje el kódolatlanul a rotált naplókat:
# mail rgazda@gyakranolvasom.hu
# Ezt kihagyjuk, mert egy script segítségével kódolva fogjuk küldeni a naplókat.
# Ha túlságosan kényelmetlennek találja az olvasó a dekódolást, és nem fél attól,
# hogy a naplóit elolvassák, akkor válassza a hagyományos módszert.

# mivel ezt a két fontos fájlt egyik csomag sem birtokolja ezért a részletes
# beállításait it közöljük (lokális)
/var/log/wtmp {
    weekly
```

```

    create 0664 root utmp
    rotate 2
}
/var/log/btmp {
    missingok
    weekly
    create 0664 root utmp
    rotate 2
}

#Ezt a részt átveszem a /etc/logrotate.d/syslog-ng fájlból és kiegészítettem.
# A syslog-ng-t újra kell indítani a rotálás után.
/var/log/syslog
{
    postrotate
        /etc/init.d/syslog-ng reload >/dev/null
        # mivel a syslog fájl mindig telik, ez a script mindig le fog futni
        # itt elindítjuk a titkosító scriptünket:
        /bin/bash /root/logtit.sh
    endscrip
}
# ha ezt nem tesszük meg, a syslog továbbra is a törölt fájlba fogja küldeni
# az információkat és azok így elvesznek, továbbá logikai lemezhibát okozhatnak.
# Frissítések során ne engedjük ezt a fájlt felülrni!
```

Ezután `rm -f /etc/logrotate.d/syslog-ng`

`touch /etc/logrotate.d/syslog-ng`

Frissítések során ne engedjük ezt a fájlt felülrni!

Ez a kis *shell-script* kódolja és postázza a rotált log-okat. Mentsük ezt a fájlt `/root/logtit.sh` néven.

```

PATH=$PATH:/bin:/usr/bin          # Beállítjuk az elérési útvonalakat.
umask 077                          # Csak a root tudja olvasni a keletkező fájlokat
cd /var/log                        # Belépünk
rm -f naplok.tar.gz 2>&1 /dev/null  # Ha esetleg lenne még itt ilyen fájl, töröljük
DATUM=`date +%Y.%m.%d-%H.%M`      # A mai dátum
tar -czf naplok.tar.gz *.0*        # Minden épp most rotált fájlt elcsomagoljuk,
cat naplok.tar.gz | gpg -e -a -q -r rgazda@gyakranolvasom.hu \
| mail -s naplorotalas-$HOSTNAME-$DATUM rgazda@gyakranolvasom.hu
rm -f naplok.tar.gz # # titkosítjuk és postázzuk. Végül töröljük az átmeneti fájlt.
```

2.8.3 A Web-szerver naplófájljai

A `cronolog` program a hosszú Web-szerver naplófájlokat szét tudja szedni adott minta szerint. Ez pl. akkor előnyös, ha a log-okat megőrizzük és eltároljuk kompakt lemezen napi bontásban. A program hátránya, hogy minden naplófájlhoz külön folyamatot indít, ezért nagy forgalmú helyeken nem ajánlott használni, ugyanis nem utólag, hanem folyamatosan válogatja szét az érkező üzeneteket. Ha beírjuk ezt a sort a `httpd.conf`-ba:

```
CustomLog „/usr/sbin/cronolog /var/log/apache-ssl/access.%Y-%m-%d.log” combined
```

akkor ilyenforma fájlokat kapunk: `access.2000-05-13.log` Minden log-fájlt hasonlóképp kezelhetünk.

2.8.4 A napló automatikus ellenőrzése

A `logcheck` program nagyon hasznos lehet számunkra. Az időzítő (`cron`) indítja el időközönként. Ez a program leellenőrzi a rendszer naplófájljait az előre megadott szűrők szerint (melyeket persze tesztre szabhatunk) és az eredményt elküldi a kért e-mail címre. Ha a szűrők megfelelnek nekünk (`/etc/logcheck/*`), akkor semmi más tennivalónk nincs vele, mint az `/etc/logcheck/logcheck.conf` fájlba beírni ezt: `SENDMAILTO=rgazda@gyarkranolvasom.hu`. Ekkor a betörési kísérletekről, biztonságot veszélyeztethető és / vagy szokatlan eseményről értesít a rendszer (mint pl. valaki `root-ra su-zott`.) Ha esetleg olyan tevékenységet is ebbe a kategóriába vesz, melyet nem tartunk veszélyesnek, akkor olvassuk el a dokumentációt és hangoljuk be a szűrőket igényeink szerint.

A probléma a módszerrel az, hogy egyrészt kódolatlanul küldi el a levelet, másrészt pedig nem tömöríti le, és ekkor sokáig tarthat a levél letöltése. Ha szeretnénk ezeket a problémákat megoldani, akkor hajtsuk végre a következő lépéseket:

1. Másoljuk az eredeti programot le két példányba:

```
cp /usr/sbin/logcheck.sh /root/logcheck.sh.orig
cp /usr/sbin/logcheck.sh /root/
```

2. Szerkesszük át a fájlt: pl. `joe /root/logcheck.sh`

```
[...]
# If there are results, mail them to sysadmin

if [ "$ATTACK" -eq 1 ]; then
    # Ha vészriadó van, nem kódoljuk le a levelet, hogy gyorsan el tudjuk olvasni
    cat $TMPDIR/checkreport.$$ | \
    $MAIL -s "$HOSTNAME $DATE A RENDSZER OSTROM ALATT! ACTIVE SYSTEM ATTACK!" $SYSADMIN
elif [ "$FOUND" -eq 1 ]; then
    # Viszont, ha csak valami gyanúsít talált, akkor azt kódoljuk.
    cat $TMPDIR/checkreport.$$ | /bin/gzip - | \
    /usr/bin/gpg -e -a -q -r $SYSADMIN | \
    $MAIL -s "$HOSTNAME $DATE rendszerellenőrzés - system check" $SYSADMIN
fi
[...]
```

A `$SYSADMIN` változó megegyezik azzal az e-mail címmel, melyet a konfigurációs fájlba írtunk. Ha ehhez az e-mail címhez nincs publikus kulcsunk (a `root` felhasználónak!), akkor hibaüzenettel le fog állni a kódolás.

3. Másoljuk vissza az átjavított fájlt: `cp /root/logcheck.sh /usr/sbin/`

Ezzel kész is. Már csak arra kell figyelniünk, hogy a programot frissüléskor ne írjuk felül, vagy tegyük „hold” állapotba, ekkor nem fog frissülni.

Az üzenetek kicsomagolásához is használhatunk egy kis script-et (írjunk egyet). A kedvenc levélolvasó programunkból mentsük ki a levél tartalmát egy fájlba, pl. `kodolt.txt` néven. Ez a művelet interaktivitást igényel, meg kell adnunk a visszakódoláshoz szükséges jelszónkat.

```
cat kodolt.txt | gpg -d | gunzip - > visszakodolt.txt
less visszakodolt.txt
```

2.9 A Web-szerver statisztikái

A `webalizer` programmal szép grafikákkal tűzdelt HTML-es statisztikákat készíthetünk a Web-szerverünk forgalmáról, melyeket azonnal közzé is tesz a Weben (persze csak ha akarjuk). Először is szerkesszük meg az `/etc/webalizer.conf` fájlt:

```
LogFile /var/log/apache-ssl/szormegyar-access.log
HostName www.szormegyar.hu
OutputDir /var/www/szormegyar/webstat
ReportTitle Jelentés
Quiet yes #ha cron-ból futtatjuk, akkor ne szövegeljen
```

A többi beállítás általában jó, de azért olvassuk végig. Ajánlatos az `access.conf`-ban azonosításhoz kötni a `/var/www/szormegyar/webstat` könyvtár tartalmát, hogy csak az illetékes személyek tekinthessék azt meg. A program kimenetét nagymértékben testre szabhatjuk. Olvassuk el a dokumentációt. A különböző virtuális szerverek forgalmát külön mérhetjük (ekkor a log formátumát is meg kell változtatni). Létrehozhatunk külön konfigurációs fájlokat minden egyes virtuális szervernek (külön a HTTPS forgalomnak is). Ezekre bővebben nem térek ki.

A programot futtathatjuk az időzítőből is, megfelelően kell csupán paraméterezni. Ez esetben a rendszer magától frissíti pl. naponta, vagy óránként a statisztikáit. Érdeemes belemélyedni, ha szükségünk van a forgalom mérésére.

A `webalizer` ehhez hasonló grafikákat készít:

18. kép - Webalizer statisztika

Ha az időzítőből szeretnénk futtatni, tegyünk egy bejegyzést. `/etc/crontab`:

```
58 06 * * * www-data webalizer >/dev/null
```

Ha külön szeretnénk mérni az egyes virtuális szerverek forgalmát, készítsünk külön konfigurációs fájlt és indítsuk el mindegyikkel egyszer-egyszer.¹⁶⁹ Ahhoz, hogy magyarul készítse el a statisztikákat, újra kell fordítani.

A másik kitűnő eszköz az `analog` programcsomag. Szintén HTML-es kimenete van, néhány grafikával megtűzdelve. Itt is érdemes elolvasni a dokumentációt, mely HTML-es formátumú. Itt kiderül milyen sok platformon életképes a program, továbbá megtanulhatjuk a használatát és a szintaxisát. Mivel szorosan nem kapcsolódik a témához, nem részletezem. Ez a program is automatizálható az időzítővel. Előnye, hogy teljesen magyar nyelvű jelentést is tud készíteni, ha beírjuk ezt a sort az `/etc/analog.conf`-ba:

```
LANGFILE /usr/lib/analog/lang/huh.lng
```

Ha szükségünk van az oldalon megjelenő látogató-számlálóra, telepítsük a `wwwcount` csomagot, mely (teljes dokumentációval ellátott) számláló program. Beállítását itt nem részletezem.

2.10 Az „upsd” beállítása

Mivel erősen függ a hardvertől az, hogy melyik csomagot fogjuk használni, ezért ebbe a témába sem merülhetek el teljességgel.

¹⁶⁹ Létezik egy sokkal elegánsabb megoldás is. Egy Perl script segítségével tetszőleges számú virtuális szerverről tudunk statisztikákat készíttetni. Herczeg Ferenc hercy@externet.hu megoldása letölthető: <http://w3.externet.hu/~narancs1/webalizerlog.pl>

A példában szereplő géphez egy *APC Smart –BackUPSPro PnP* eszközt vásároltunk, melyet az első soros portra illesztettünk (`/dev/ttyS0`). Ennek megfelelően az `apcupsd` programot telepítettem.

Szerkesszük meg az `/etc/apcupsd.conf`-ot. Adjuk meg az UPS típusát, a kábel típusát, és a portot. Mivel csak ezt az egy gépet szolgálja ki a tápegység, nincs szükség hálózati beállításokra. A tápegység állapota akár Web-es felületen is lekérdezhető. Mindenképp végezzük el a finomhangolást a dokumentáció elolvasása után. Ha lehet, állítsuk be a démont úgy, hogy hiba esetén e-mail-t küldjön címünkre.

2.11 A biztonsági mentés időzítése

Én a mentéshez a `kbackup` csomagot választottam. Egyrészt jól dokumentált (külön csomag), másrészt egyszerűen használható mind szalagos, mind floppy-s mentésekhez. Ha telepítettük a `dialog` csomagot, akkor a `make menuconfig`-hoz hasonló könnyen kezelhető menüs programot kapunk. A különböző partíciókat érdemes külön-külön menteni. Minden mentéshez egyedi beállításokat menthetünk el, melyeket később újra fel lehet használni. A *Schedule* menüpontban beállíthatjuk az automatikus időzítéseket is. Olvassuk el a dokumentációt és állítsunk be mindent a tervek szerint. Készítsünk próbamentéseket és visszaállításokat egy üres partícióra.

2.12 A szükséges felhasználók/csoportok létrehozása és a lemezkvóták beállítása

A felhasználói számlák kreálásának alapszabályai az `/etc/adduser.conf`¹⁷⁰ fájlban találhatóak. Tudnunk kell, hogy alapbeállításban minden felhasználó létrehozásakor egy azonos nevű csoport is keletkezik, melybe az illető természetesen bele fog tartozni. Ez sok felhasználó esetén nem jó eljárás, de esetünkben még megfelel. (Ha nekünk ez mégse felelne meg, akkor a fenti állományban állítsuk a `USERGROUPS=yes` értékét `no`-ra.) Az az általános eljárás, hogy létrehoznak egy általános `users` csoportot. Ennek a jogait és korlátait (kvótáit) minden felhasználó örökli keletkezésakor. Ezen túlmenően specifikus csoportok is létrehozhatóak, melybe később fel lehet venni a megfelelő illetőket.

Ha már létrehoztunk egy sablont és annak beállítottuk a kvótáit, akkor a `QUOTAUSER="sablon"` sorral ezt kiterjeszthetjük az összes ezután keletkező felhasználóra.

Minden olyan fájlt (pl. a „.”-al kezdődő „*rc*” fájlokat¹⁷¹) helyezzünk el az `/etc/skel` könyvtárban, melyet az új felhasználók meg kell, hogy kapjanak. (A programok alapbeállításait.)

¹⁷⁰ Olvassuk el a manuál oldalát: `man adduser.conf`

¹⁷¹ Ezek tartalmazzák a különböző programok felhasználói szintű beállításait.

A kvóták beállításához először olvassuk el a dokumentációt¹⁷², majd hozzunk létre egy sablonfelhasználót. Ennek a kvótáit állítsuk be a kívánt értékre az `edquota -u sablon` (Itt a program behívja a `vi` szövegszerkesztőt, hacsak az `EDITOR` környezeti változóban ezt át nem definiáltuk. Ha nem találja a szövegszerkesztőt, akkor hibaüzenettel leáll.) (1 *block*= 1024 *bytes*)

```
Quotas for user sablon
/dev/hda6: blocks in use: 4, limits (soft = 2000, hard = 4000)
        inodes in use: 5, limits (soft = 500, hard = 1000)
```

Később külön-külön is szabályozhatjuk a kvóták határait az egyes felhasználókra. Járjunk el hasonlóképp a többi kvótát igénylő partíció esetén is.

A tervek szerint hozzuk létre a felhasználókat az `adduser fnév` paranccsal¹⁷³. Kérjünk mindenkitől egy jelszót, amit most megadhatunk, és amelyet nekik rögtön meg is kell változtatniuk a `passwd` paranccsal (vagy találjunk ki mi magunk valamit, esetleg a jelszógeneráló programokat használjuk fel). Felhasználót a `userdel fnév` paranccsal törölhetünk.

Csoportokat az `addgroup` paranccsal hozhatunk létre. Felhasználót a csoportba `adduser fnév csoport` paranccsal adhatunk. Egy adott csoport tagjait kilistázzhatjuk a `members csoport` paranccsal (ha telepítettük).

Egy részletesebb leírást találhatunk a felhasználók / csoportok létrehozásának technikájáról itt¹⁷⁴.

2.13 A „/etc/fstab” és az „init script”-ek beállítása

Ha úgy véljük, hogy a rendszer kész, szerkesszük meg az `/etc/fstab` fájlt, hogy a különböző partíciókra korlátozásokat vezessünk be a terveink szerint. Először is mentsük el az eredetit `fstab.rw` néven: `cp /etc/fstab /etc/fstab.rw` A kedvenc szövegszerkesztőnkkel pedig tegyük meg az alábbi változtatásokat:

```
# /etc/fstab: statikus file-rendszer információk. # #
/dev/hda3 /          ext2    defaults                    0 1
/dev/hda2 none             swap    sw                        0 0
proc      /proc          proc    defaults                  0 0
# Vedd ki a kommentből a következő sort, ha 2.2.x vagy újabb kernelnél
# UNIX98-szerű pty kezelést szeretnél
none      /dev/pts        devpts  gid=5,mode=620            0 0
#/dev/fd0 /floppy         auto    defaults,user,noauto      0 0
#/dev/cdrom /cdrom          iso9660 defaults,ro,user,noauto    0 0
/dev/hda1 /boot           ext2    ro,nosuid,noexec,nodev,defaults 0 2
```

¹⁷² [/usr/doc/quota/html/quota.html](http://usr/doc/quota/html/quota.html)

¹⁷³ `man adduser`

¹⁷⁴ <http://www.vmg.sulinet.hu/vmg/home/szamtech/linux/node221.htm> Itt található egy bővebb leírás.

/dev/hda5	/usr	ext2	ro,nodev,defaults	0 2
/dev/hda6	/home	ext2	defaults,nosuid,nodev,noexec, usrquota	0 2
/dev/hda7	/tmp	ext2	nosuid,noexec,nodev,defaults, usrquota	0 2
# /var noexec esetén nem futnak pre és postinst scriptek csomagok telepítésekor!				
/dev/hda8	/var	ext2	nosuid,noexec,nodev,defaults	0 2
/dev/hda9	/var/www	ext2	nosuid,noexec,nodev,defaults	0 2

Ha a gyöker partíciót *ro*-ban szeretnénk használni, először bele kell nyúlnunk az *init* fájllokba, mivel azok oda írni akarnak. Bizonyos indító script-eket át kell szerkeszteni, és módosítani kell minden */etc/**-ra való fájlműveletet */var/**-ra

A következő fájlok érintettek: */etc/motd*, */etc/nologin*, */etc/nologin.boot*, */etc/mtab*, */etc/adjtime*, */dev/MAKEDEV*, */dev/log* és ezeket kell módosítanunk: */etc/init.d/checkroot.sh*, */etc/init.d/bootmisc.sh*, */etc/init.d/rmnologin*, */etc/init.d/hwclock.sh*, */etc/init.d/makedev*, */etc/syslog-ng/syslog-ng.conf*, */etc/default/rcS*

Az */etc/mtab* fájl tárolja azt a listát, hogy melyik partíció hogyan van felfűzve a rendszerbe. Szerencsére a 2.2-es kernelektől kezdve létezik egy */proc/mounts* fájl, mely ugyanezeket az információkat tartalmazza¹⁷⁵. Így nem lesz szükség az *mtab*-ra.

```
rm -f /etc/mtab
ln -s /var/mtab /proc/mounts
/etc/init.d/checkroot.sh:
```

```
[...]
# elméletileg ez csak akkor indul el, ha a / rw-ben van. Azért szedjük ki, mert
# elpiszkálja a szimbolikus linkünket.
if [ "$mode" = rw ]
then
    rm -f /var/nologin #/etc/mtab~
    #: > /etc/mtab
[...]
```

A *nologin* fájl arra szolgál, hogy amíg a rendszer indulási, vagy leállási folyamatban van, ne lehessen bejelentkezni. A fájlt ha áthelyezzük, a programoknak ugyanúgy észre kell vennie:

```
ln -s /var/nologin /etc/nologin
ln -s /var/nologin.boot /etc/nologin.boot
```

A *nologin* problémát kiküszöbölhetjük úgy is, ha az */etc/default/rcS* fájlba ezt írjuk: (Ez viszont nem javasolt.)

```
DELAYLOGIN=no
```

A *motd* (*Message of the day*, a napi üzenet) problémáját így oldjuk meg:

```
mv /etc/motd /var; ln -s /var/motd /etc/motd
```

¹⁷⁵ Néhány információt így viszont nem fogunk látni! Pl. GUID, umask, stb.

```
/etc/init.d/bootmisc.sh
```

```
[...]
if [ "$DELAYLOGIN" = yes ]
then
    echo "System bootup in progress - please wait" > /var/nologin
    cp /var/nologin /var/nologin.boot
fi
[...]
# Update /etc/motd.
#
if [ "$EDITMOTD" != no ]
then
    uname -a > /var/motd.tmp
    sed 1d /var/motd >> /var/motd.tmp
    mv /var/motd.tmp /var/motd
fi
[...]
```

```
/etc/init.d/rmnologin:
```

```
[...]
if [ -f /var/nologin.boot ]
then
    rm -f /var/nologin /var/nologin.boot
fi
[...]
```

Vagy a *motd* problémát kiküszöbölhetjük úgy is, ha az */etc/default/rcS* fájlba ezt írjuk:

```
EDITMOTD=no
```

Az */etc/adjtime* fájl tárolja a hardver idő-beállításait.

```
mv /etc/adjtime /var/adjtime
ln -s /var/adjtime /etc/adjtime
/etc/init.d/hwclock.sh:
```

```
[...]
start)
    if [ ! -f /var/adjtime ]
    then
        echo "0.0 0 0.0" > /var/adjtime
    fi
[...]
```

Az */etc/init.d/makedev* script feladata biztosítani azt, hogy a */dev/MAKEDEV* egy szimbolikus link legyen az */sbin/MAKEDEV* fájlra. Erre nincs szükségünk, ezért írjunk *exit 0*-át a script első sorába.

A következő probléma az, hogy a *syslog-ng* induláskor megnyitná a */dev/log* fájlt. Persze ez esetünkben nem lehetséges.

```
/etc/init.d/syslog-ng stop
```

```
mv /dev/log /dev/log1; ln -s /var/run/log /dev/log;  
/etc/syslog-ng/syslog-ng.conf:  
source src { unix-stream("/var/run/log"); internal(); };  
/etc/init.d/syslog-ng start
```

Ha mindent elrendeztünk és leellenőriztünk, adjunk ki egy `sync` parancsot, majd egy `reboot -ot`¹⁷⁶. Újrainduláskor figyeljük meg a hibaüzeneteket (ha vannak). A modulfüggőségeket nem fogja tudni a rendszer újra elkészíteni, de ez nem is baj. A *shift-pageup* billentyű kombinációval visszalapozhatunk a képernyőn. Ha később változtatni akarunk valamit, a `mount / -o remount,rw` paranccsal újra felfűzhetjük az adott partíciót írható módban.

A fenti fájlok letölthetőek: <http://w3.exnet.hu/~narancs1/ro-root-0.1.tgz>

2.14 A „tripwire” program beállítása és használata

A `tripwire` program egy fájlintegritás ellenőrző segédeszköz. Miután minden működik és teljesen készretettük a rendszert, beleértve az összes beállítást és tesztet, készítünk egy adatbázist, mely a fontos fájlok és könyvtárak méretét és jogosultságait eltárolja MD5-ös algoritmust használva. A `tripwire-t` később az időzítőből futtatva összehasonlíthatjuk a fájlok állapotát az adatbázisban tároltakkal. Ekkor a program levelet küld nekünk, hogy változott-e valami. Ha változott, akkor vagy kompromittálták a rendszert, vagy mi magunk csinálhattunk valamit. Az adatbázis részeit igény szerint frissíthetjük is.

A dokumentáció és a manuál oldal¹⁷⁷ elolvasása után szerkesszük meg az `/etc/tripwire/tw.config` fájlt.

```
[...]
/                R          # ezek a partíciók jobb esetben ro-ban vannak
/usr             R          # tehát ezekről készítünk lenyomatot, mert nem
/boot           R          # szabad, hogy változzanak
/dev            @@DEVSEARCH # speciális keresést kap
=/home          # ezek egy minimális tesztet kérünk
=/tmp           @@TMPSEARCH
=/var/tmp       @@TMPSEARCH
!/mnt           # ezeket nem kell ellenőrizni
!/floppy
!/cdrom
!/var
!/var/www
!/usr/doc        # ezeket a fájlokat nem szükséges ellenőrizni
!/usr/share/doc  # de ha valakinek ez is fontos, akkor vegye ki innen
!/usr/dict       # a "!" jelet és ellenőriztesse.
!/usr/info
```

¹⁷⁶ Egyébként nem szükséges újraindítani a rendszert. Elég lenne kézzel `mount -o remount,opciók` paranccsal egyenként újra felfűzni a partíciókat. Az újraindítás csak az ellenőrzés kedvéért történik.

¹⁷⁷ man tripwire, man tw.config

```
!/usr/man          # a többi beállítást hagyjuk meg.  
[...]
```

Ahhoz, hogy az változásokat nyomonkövessük, a rendszer időzítoje mindennap lefuttatja az ellenőrzést. Szerkesszük meg az `/etc/cron.daily/tripwire` fájlt:

```
#!/bin/sh  
cd /var  
# hol az adatbázis? Lehet tömörített is.  
DATABASE="/root/databases/tw.db_`hostname`"  
DATABASEGZIP="/root/databases/tw.db_`hostname`.gz"  
LOG=/var/log/tripwire  
# hova köldjük a levelet?  
MAILTO=rgazda@gyakranolvasom.hu  
[...]  
# if the temporary file is empty do not send mail  
# azt akarjuk, hogy ha nincs változás ne küldjön levelet,  
[ ! -s $LOG -o -z "$MAILTO" ] && exit 0  
# ha mindig szeretnénk levelet kapni, akkor kommentezzük ki.  
# ha gondoljuk, írjuk át a levél szövegét magyarra:  
(cat <<EOF;  
This is an automated report of possible file integrity changes, generated by  
the Tripwire integrity checker.  
Ezt egy automatikusan készülő levél, mely a lehetséges fájlintegritás-változásokat  
tartalmazza, és melyet a Tripwire program készít.  
  
Changed files/directories include:  
A megváltozott fájlok / könyvtárak listája:  
EOF  
cat $LOG  
) | /usr/bin/mail -s "Fájl integritás-jelentés - File integrity report" $MAILTO
```

Ha úgy gondoljuk, hogy ezt a levelet is kódoltan szeretnénk megkapni, akkor módosítsuk az utolsó sort:

```
) | /bin/gzip - | \  
/usr/bin/gpg -e -a -q -r $MAILTO | \  
/usr/bin/mail -s "Fájl integritás-jelentés - File integrity report" $MAILTO
```

Ha nem szeretnénk a `gzip`-el bajlódni, hagyjuk ki. Nem biztos, hogy olyan hosszú levelet fog készíteni.

Ha ez kész a `tripwire -initialize` paranccsal készíthetjük el az adatbázist. Az adatbázis eredetileg a `./databases` könyvtárba kerül, ezért tanácsos a `/root` könyvtárban állnunk. Persze itt nem biztos, hogy jó helyen van. Mindenképp mentsük ki egy példányban floppy-ra (tömörítve). Ezt a lemezt helyezzük biztonságba. Ha a rendszert kompromittálták, lehet, hogy az adatbázist is elérték. Ekkor elővehetjük a lemezt és az ellenőrzést ez alapján is elvégezhetjük.

Ha később szeretnénk változtatásokat felvinni az adatbázisba, akkor az `-interactive` kapcsolóval indítsuk. Az integritás ellenőrzésekor elkészít egy új adatbázist és azt hasonlítja össze a régivel. Ezért írható könyvtárban kell indítani.

Paraméterként adjuk meg neki az adatbázist a `-d adatbázis` kapcsolóval. Frissítés után mentjük el az új adatbázist is lemezre.

Ezután végezzük el a gyökérpartíció újrafelfűzését *ro* módban.

V. Egy gyakorlati példa bemutatása

Ezt a fejezetet arra szánom, hogy az életben már működő rendszerekről kapjunk egy rövid képet.

Ezt a rövid összefoglalót Szentmarjay Tibor (tibor@naplopok.hu) kollégám küldte e-mail-ben, bemutatva az Ő Debian rendszerét. (Az adott cég nevét jogi okok miatt kihúztam.)

“Nos.

A hardver: AMD K6-2 300MHz, Aristo TX MVP alaplap, 64MB SDRAM, van benne egy 3,2GB-os és egy 13GB-os IDE-s merevlemez, ebből 6GB backup, home könyvtáraknak 9GB van adva.

- *A szervert 50 db user használja, elérésük FTP-re van korlátozva, csak 5 usernek van SSH loginra engedélye. telnet le van tiltva. Használható még a POP3 (most tervezem ennek a biztonsági védelmét). Hogy mennyi anyagot szolgáltat, azt most nem tudom megmondani, de az összesített heti access.log az apache-ből 450MB felett van.*
- *Apache httpd.conf (ami nem default): MaxClients 200. A többi is fel van emelve egy kicsit. 46 virtualhost fut az apache alatt, ez fizikailag nem a httpd.conf-ban vannak, hanem egy Include-dal vannak berakva, és egy vhosts.conf fájlban találhatóak.*
- *A szerver melleleg alkalmas WAP megjelenítésére is, ehhez csak pár sor kellett a mime.types-ba.*
- *PHP3 van installálva, mindenkinek engedélyezve van. PERL 5 van még a gépen. Ez csak bizonyos felhasználóknak van engedélyezve, külön CGI-BIN könyvtárral. “Apache suid” kiegészítés nincs feltelepítve.*
- *Backup naponta van, a backupolt tar.gzip fájlok mérete 500MB körül szokott lenni.*
- *A hálókártya eredetileg egy Planet 10/100-as volt (100-as hálón vagyunk az ****-nél) Realtek chipsettel, csak sajna a RealTek érzékelny 100Mbit esetén a hálózati fizikai hibákra és túlterhelésre, így párszor eldobta magát (nem túl soxor). Erre akartunk egy szkriptet írni, csak downolni kellett volna az ifconfigot és utána ujraindítani, de inkább vettünk egy Digital (Tulip) chipsetes 10/100-as Planetet. 3Comot akartunk, de az túl drága volt, és ezzel a kártyával most már hónapok óta semmi gond sem volt.*
- *A gép amúgy az ***** cég 100Mbites switch-én van, szünetmentes táp és légkondi biztosítva. Nincs firewall előtte.*
- *SSL most épp nincs, teszt szinten volt fent apache-ssl, de most sima apache van, most lesz mod_ssl installálva, mivel lesz rajta hitelkártyás fizetés, és most fejlesztjük a webes adminisztrációt is. A usereknek is most készül webes karbantartás (levélforward, kulcsszóválogztatás stb.) Statisztikák a usereknek a következő szoftverekkel vannak biztosítva: easystat (ez magyar) és analog (ez most fordítjuk magyarra).*

- *Adatbázis elérés csak akkor engedett, ha a user külön kéri, az adatbázist. Mi hozzuk létre, azon belül ő azt csinál, amit akar. Ehhez is lenne webes adminisztráció, de még nem volt rá igény, mindenki PHP3-ból és Perlből kezeli.*
- *Mivel több domaint is tárolunk, felmerült a domainhez illeszkedő emailcímek biztosítása is. Ez a qmail + vchkpwd + vpopmail nevű cuccokkal lett megoldva. Az utóbbi kettőt a www.inter7.com címen megtalálod. Ezekkel felhasználó létrehozása nélkül lehet POP-os postaládákat létrehozni, levlistákat is. Webes kezelőfelülete is van*
- *Az emailezéshez van webmailünk, ez is letölthető az inter7.com-ról, neve vsqwebmail.*
- *FTP daemon wu-ftpd. Mindenki csak a saját home-jában barangolhat.*

[...]

Ui:

Mem: 63428K av, 61640K used, 1788K free, 250120K shrd, 5644K buff

Swap: 130748K av, 7632K used, 123116K free 16572K cached

CPU states: 2.7% user, 4.6% system, 92.5% nice, 0.0% idle

12:02pm up 40 days, 10:46, 2 users, load average: 2.04, 2.10, 2.11

134 processes: 130 sleeping, 4 running, 0 zombie, 0 stopped

a load azért van 2-n, mert fut egy distributed.net kliens.”

VI. A jövő

Hogy mit hoz a jövő azt a számítástechnikában igen nehéz megjósolni. Egyesek szerint a Linux világalomra tör, mások szerint végül mindenki át fog térni majd a FreeBSD-re (vagy valamelyik más BSD variánsra). Néhányan pedig a szabadszoftver-mozgalom teljes bukását és a kommersziális szoftverek monopolista-centralista Orwell-i rémképét vetítik felénk. Ma még nem tudhatjuk.

Saját véleményem az, hogy a Linux és a szabad szoftverek széles körben elterjednek, és használatban lesznek az informatika minden területén, többek között az otthoni és kisvállalkozói asztalon – a másik végletben a nagyvállalati szerverközpontokban is.

1. Az új kernel és a khttpd

A nem is olyan távoli jövő Linux kezele akár kávé is tud majd nekünk főzni és felhív telefonon, ha baj van a számítógép áramellátásával. (Már létezik olyan kézigép, amely egyben mobiltelefon, határidőnapló, email-kliens, Web-böngésző, diktafon és Linux-ot futtat.)

Viccet félretéve, a most megjelenő 2.4-es kernelsorozat rengeteg újítást és sok új eszközvezérlőt tartalmaz. Ezzel a rendszermaggal szeretnének betörni az „Enterprise” (nagyvállalati) piacra.¹⁷⁸

Néhány paraméter és újdonság:¹⁷⁹

- teljesen új erőforrás-menedzsment alrendszer (*plug-and-play*¹⁸⁰)
- automatikus és kernelszintű ISA-PnP
- új VFS¹⁸¹ és *cache*¹⁸²-rendszer
- 4.2 billió felhasználó (és csoport) használhatja egyidejűleg a rendszert
- több mint 4 GB memória egy gépben (akár 64GB is lehet)
- akár 16 db Ethernet-kártya egy gépben
- akár 10 db IDE vezérlő egy gépben
- a fájlrendszer réteg újraírva, több mint 8 processzoros rendszerekhez optimalizálva¹⁸³
- új SMP ütemező, jobb multiprocesszoros teljesítmény, jobban skálázható
- NFSv3 implementáció (*Network FileSystem version 3*)

¹⁷⁸ Sajnos az ezt igazán lehetővé tevő JFS változatok (Naplózó Fájlrendszer) egyike sem része még a hivatalos 2.4-es sorozatnak, csupán foltként hozzáférhető.

¹⁷⁹ Az adatokat a következő cikkből merítettem: Joe Pranevich: The wonderful world of Linux 2.4, <http://www.linuxtoday.com/stories/19356.html>

¹⁸⁰ Automatikus, szoftveres hardver-beállítás.

¹⁸¹ Virtual File System, virtuális fájl-rendszer

¹⁸² Memóriában lévő lemeztár-gyorsító

¹⁸³ Ez persze nem azt jelenti, hogy egy processzossal lassabb, hanem azt, hogy jobban ki tudja használni a több processzort, mert a legtöbb processzorigényes párhuzamosítható műveletet újraírták és optimalizálták.

- IA64 és Crusoe processzorok támogatása
- DevFS (*Device FileSystem*) – a hardvereszközök ezután nem a lemezen helyezkednek el a `/dev` könyvtárban, hanem a kernel kezeli azokat. Amikor betöltődik a kernelbe az eszközezerlő és az megtalálja a hardvert, akkor jelenik meg a hozzá tartozó eszközjelölő. (Ez persze a terjesztések átalakítását is igényli.)
- I2O¹⁸⁴ támogatás
- PCMCIA és PC Card támogatás integrálása a kernelbe
- Teljes USB¹⁸⁵ támogatás
- FireWire támogatás (nagysebességű eszközökhöz)
- LVM (*Logical Volume Manager*) új partícionálási eljárás
- „wake one” – pl. egy Web-szerver esetén ha kérés érkezik a hálózatról, akkor csak egy folyamatot „kelt föl” a kernel, nem többet, mert akkor
 1. azok versengenek a kapcsolat kiszolgálásáért
 2. úgyis csak egy kaphatja meg a kérést, így a többi feleslegesen foglalta le az erőforrásokat.Ez nagyban javíthatja a Web-szerverek teljesítményét.
- Teljesen újraírt és párhuzamosított hálózati réteg
- Új csomagszűrő alrendszer (*NetFilter*, *NAT Network Address Translation*)
- Új PPP alrendszer, új ISDN vezérlők
- és még sok más...

kHTTPd: kernelszintű HTTP démon: ez egy ún. „közvetlen” hálózati kiszolgáló, egy nem-kernelszintű Web-szerver szoftverrel kombinálva nagyban meggyorsíthatja a statikus tartalmak kiszolgálását (Értsd: statikus HTML oldalak, képek, programfájlok, stb.) Ez a rendszer nem képes pl. CGI-t futtatni és dinamikus oldalakat generálni. Ha ki akarjuk próbálni, mindenképp fordítsuk modulba.

Tapasztalataim szerint ez a kernelszintű Web-gyorsítás igen hatásos. Maga az egész rendszermag sokkal gyorsabb és ténylegesen hatékonyabb a 2.2-es sorozatnál. Ha nagyon számít a sebesség, akkor mindenképp érdemes megpróbálkozni vele. Egyetlen nagy hátránya, hogy a *virtual hosting*-ot még nem ismeri, csak egyetlen *documentroot* áll rendelkezésre.

Fontos!

Az új kernelnek teljesen más a csomagszűrő rendszere, az *ipchains* ugyan még működik egy ún. „*wrapper*”¹⁸⁶ modul segítségével. Nézzünk szét a kernel konfigurációs *Netfilter* szekciójában és tegyük mindent modulba, amire szükségünk lehet.

¹⁸⁴ Intelligent Input/Output: segítségével operációs rendszer független eszközezerlők írhatóak.

¹⁸⁵ Universal Serial Bus: Új, nagyobb sebességű soros-port szabvány.

¹⁸⁶ Itt kb. annyit jelent magyarul, hogy közvetítő, átjátszó.

A következő indítóprogramot helyezzük el az /etc/init.d/ könyvtárban és nevezzük el khttpd-nek:

```
#!/bin/sh
#-----Beállítások
clientport=80                #Az igazi webszerver port-ja
serverport=8080              #A kernel HTTPD figyelési portja (ha élesben akarjuk
# használni, akkor cseréljük meg a kettőt. SSL-es kapcsolatokat nem kezel.
documentroot=/var/www        #Az adott webszerver dokumentum-gyökérkönyvtára
threads=2                    #hány szál induljon el, processzoronként egy javasolt.
din1=php3                    # milyen kiterjesztése van a dinamikus oldalnak
din2=shtml
din3=cgi
#-----Indítófájl
NAME="khttpd"
DESC="Kernel httpd accelator daemon"
set -e
case "$1" in
    start)
        echo -n "Starting $DESC: "
        modprobe khttpd #itt töltjük be a kernelmodult
        echo $clientport > /proc/sys/net/khttpd/clientport
        echo $serverport > /proc/sys/net/khttpd/serverport
        echo $documentroot > /proc/sys/net/khttpd/documentroot
        echo $threads > /proc/sys/net/khttpd/threads
        echo $din1 > /proc/sys/net/khttpd/dynamic
        echo $din2 >> /proc/sys/net/khttpd/dynamic
        echo $din3 >> /proc/sys/net/khttpd/dynamic
        echo 1 > /proc/sys/net/khttpd/start
        #a fenti sorokkal a khttpd konfigurációját írtuk be a /proc fs-be
        echo "$NAME."          ;;
    stop)
        echo -n "Stopping $DESC: "      # leállítás
        echo 1 > /proc/sys/net/khttpd/stop
        echo 1 > /proc/sys/net/khttpd/unload
        echo "$NAME."          ;;
    restart|force-reload)
        echo -n "Restarting $DESC: "    # újraindítás
        echo 1 > /proc/sys/net/khttpd/stop
        sleep 1
        echo 1 > /proc/sys/net/khttpd/start
        echo "$NAME."          ;;
    *)
        N=/etc/init.d/$NAME
        # echo "Usage: $N {start|stop|restart|reload|force-reload}" >&2
        echo "Usage: $N {start|stop|restart|force-reload}" >&2
        exit 1                  ;;
esac
exit 0
```

2. Az új Debian

Az új Debian változat kódneve „Woody”. Még keveset tudni róla. Most kezdték csak fejleszteni és a programozók szerint aktívan fejlesztik. Valószínűleg 2.4-es kernelre fog épülni. Tartalmazni fogja a 2.0-s Apache-ot, a végleges PHP4-et és egy sereg más friss programot. Valószínűleg 2001 második felében lesz kész.

3. Az új Apache

Az új Apache szerver a 2.0-s sorozat lesz. Főleg a skálázhatóság és a sebesség növelése volt a cél a fejlesztésnél. Jelenleg a 2.0a3 (vagyis a 2.0-s verzió harmadik alfája) érhető el. Már sokan letöltötték és kipróbálták. Számunkra ez a verzió addig nem lehet aktuális, amíg a Debian stabil változata nem ezt fogja tartalmazni, és szervereknél nem ajánlatos alkalmazni.

Ezek az új tulajdonságok lesznek benne:

Mag bővítések:

- *Unix Threading* (többszálúsítás). Olyan UN*X alapú rendszereken, melyek POSIX kompatibilis többszálúságot nyújtanak, az Apache képes futni hibrid többszálú és több processzű üzemmódban. Ez a skálázhatóságot növeli.
- Új fordítási mechanizmus. *Autoconf* és *libtool* alapú fordítás előtti konfigurálás.¹⁸⁷
- Multiprotokoll támogatás.
- Nem UN*X alapú rendszerek jobb támogatása.
- Új API. Az API jelentősen megváltozott az 1.3-as változathoz képest. Új modul-rendező eljárás.

Sajnos erről sem lehet még sokat tudni, kevés információ van fenn a hivatalos oldalakon, még erősen fejlesztés alatt áll.

4. PHP4

Ahogy a PHP egyre népszerűbb és elterjedtebb lett, egyre többféle platformon szeretnék alkalmazni. Az új változat sokkal skálázhatóbb, nagyobb teljesítményű és sokkal több platformon fut, mint elődje. A fordítási eljárást is leegyszerűsítették. A rendszer motorját teljesen újraírták.

Újdonságok:

- A „Zend” motor. Ez a PHP-nek egy teljesen újraírt rendszermagja. Nagy teljesítménybeli gyorsulások várhatóak tőle. Továbbá tartalmaz új nyelvi elemeket, mint pl. bővített objektum-kezelés, új változó típusok.

¹⁸⁷ Ezekkel a metódusokkal egyszerűbbé válik majd a fordítás.

- Szerver Absztrakciós Réteg. Segítségével nem csak az Apache-ba integrálható könnyedén, hanem más Web-szerver szoftverekbe is.
- „*HTTP session*” natív támogatás.
- Általánosított fordítási mechanizmus UN*X típusú rendszereknél. Megkönnyíti az újabb PHP modulok dinamikus fordítását.
- Az Apache konfigurációs fájljain keresztül beállítható a PHP viselkedése is.

A PHP4 már kész, nem béta szoftver. Rengeteg előnnyel rendelkezik a PHP3-al szemben. Hátránya épp az újdonságában rejlik:

- Nem tesztelt
- Még csak kevesen értenek hozzá
- Kevés írott dokumentáció / szakkönyv kapható hozzá (vagy épp egy se)

Gondoljuk végig, kell-e nekünk az újabbik változat. Ha igen, keressünk dokumentációkat a Web-oldalon.

A Potato-ban benne van a PHP4 béta 3. A PHP4 végleges változata valószínűleg már csak a Woody-ban lesz meg.

VII. Alternatívát nyújtó programok a Debian-ban

Egy-egy adott alkalmazási, szolgáltatási területre rengeteg megvalósítás létezik. A Debian egyik előnye, hogy megadja nekünk a választási lehetőséget, mit szeretnénk használni pl. a Web-szolgáltatáshoz.

Mivel a szabad szoftverek között ugyanarra a munkafeladatra sok alternatíva van, várható, hogy a különböző implementációk teljesen más alapokból építhetnek és teljesen más célra irányulhatnak. Az eredménye ennek az, hogy az adott feladathoz a lehetőséghez képest legmegfelelőbb szoftvert választhassuk ki.

1. Alternatívák a httpd-re

Bár az Apache a legnépszerűbb és legelterjedtebb e protokoll szerver-szintű megvalósításában, mindig akadnak másként gondolkodók. Ezek az emberek és szoftverek nagyon fontosak, hiszen itt is egyfajta evolúcióról és versenyről van szó, mint az élet sok más területén. Ha az olvasónak nem nyerte el az Apache a tetszését, bátran lehet próbálkozni más eszközökkel is.

1.1 Roxen

A UN*X rendszereken a *Roxen* a leggyakrabban használt szabad Web-szerver az Apache után. *Pike* nyelven írták, ebből kifolyólag futás közben interpretálódik, ezért bizonyos esetekben lassabb, mint az Apache. Előnye viszont, hogy rengeteg szabad és kommersziális modul van hozzá, könnyen bővíthető és programozható. Böngészőn keresztül kényelmesen konfigurálható.

A *Potato*-ban jelenleg az 1.3.122-es változat található. Tudni kell, azonban, hogy a 2.0-s változat már egy komplex Web-es alkalmazás-fejlesztő szoftverrendszerbe lesz ágyazva, melyet *Roxen Platform 2.0*-ának neveznek. Ennek a motorja a *Challenger* Web-szerver 2.0 (GPL).

Tulajdonságai:

- *Java* 2 Modulok
- *Java Servlets* 2.2 támogatás
- Belső PHP4 támogatás
- XML kompatibilis RXML értelmező
- Adatbázis illesztők kommersziális szoftverekhez
- Unikód használata
- Új konfigurációs felület
- Több felhasználó különböző jogosultságokkal
- Többnyelvű felület

- Témázhatóság
- Új alternatív HTTP modul extrém nagy sebességekhez
- Új frissítő rendszer – menet közben

A Potato-ban a Roxen több csomagra van feldarabolva. Ezeket nem szeretném felsorolni, mert egyrészt túl sok van belőlük, másrészt már túlmutat a téma keretein. Bővebb információkért olvassuk el a csomagokhoz mellékelt rövid magyarázatokat és nézzük meg a rendszer Web-helyét <http://www..roxen.com>

1.2 Zope – Z Object Publishing Enviroment

A Zope egy vezető Web-es alkalmazás-szerver. Kitűnő Web-tartalom fejlesztőeszköz, főleg csoportmunkára van kiélezve. Nagyon gyorsan és könnyen lehet vele dinamikus és interaktív Web-helyeket fejleszteni. Alkalmazási területei:

- Web alapú üzleti alkalmazások készítése
- Portálok létrehozása
- Személyre szabás
- Online hírek
- *site-search*, stb.

Beépített Web-szervert és kereső motort tartalmaz. Majdnem minden UN*X típusú platformon fut, portolták (nem-un*x) kereskedelmi rendszerekre is. Támogatja a XML-RPC, DOM, és WebDAV Web-es szabványokat is. Sok kisebb komponensből áll: Internet (Web) szerver, tranzakciós objektum adatbázis, kereső motor, Web-oldal mintázó rendszer, Web-fejlesztő és karbantartó rendszer és bővítési támogatás.

A Debian-ban a Zope is több részre van szétszedve. Ezeket sem sorolom fel. A Potato-ban a 2.1.4-es változat van, bár a Web-helyükön már a 2.1.6-os is elérhető. Illesztő fellelhető mind a MySQL és PostGRES adatbázis szerverekhez is.

Bővebb információkért a csomaglistában és a <http://www..zope.org>-on keresgéljünk.

1.3 Kisebb szerverek

Meg kell említenem még három kisebb http szervert.

- Az első a *cern-httpd*. Ebből fejlődött ki később az Apache. Ezt már nem fejlesztik tovább, bár még néhány helyen használják. Kis teljesítmény, kis tudás. Már történelmi jellegű (és jelentőségű).
- *dhttpd* – Minimális, biztonságos Web-szerver. Nincs CGI-bin támogatás! Mivel nem futtat külső programokat (csak statikus tartalmat szolgáltat) nem lehet könnyen feltörni. Nincs szüksége állandó IP címre, kevés erőforrást fogyaszt. Felhasználók is futtathatják magasabb portokon. Nem kell konfigurálni.
- *boa* – pehelysúlyú, nagy teljesítményű Web-szerver. Csak egy folyamatot futtat, nem indít újakat több kérés esetén, belülről osztja szét a kéréseket. Csupán CGI futtatásakor indít új folyamatot. Főleg régebbi, kis teljesítményű

gépekhez (is) ajánlott. Olyan feladatokra, ahová a többi program túl nagy és lassú lenne.

A fentiek mind elérhetőek a Potato-ban.

2. Alternatívák a dinamikus HTML-ek generálására

Erre a célra sokkal régebb óta használják a **CGI** módszert. Ekkor a Web-szerver meghív egy külső értelmezőt vagy programot, mely legenerálja a tartalmat és az átadja a szervernek. Főképp a **Perl** nyelvi eszközöket szokták erre a célra használni. Külső program meghívása elkerülhető egy illesztő Apache modul használatával a legtöbb eredetileg CGI típusú értelmező nyelv esetén. Pl. a `libapache-mod-perl` csomag a Perl nyelvi elemek használatának az Apache-ba való integrálást segíti elő. Ezzel 400-2000%-os gyorsulást érhetünk el a hagyományos módszerrel szemben. A Perl nyelvi eszközök illesztői megszámlálhatatlan programhoz, programkönyvtárhoz léteznek. Többek között adatbázis szerverek, grafikai könyvtárak, az XML. A Potato-ban 197 csomag nevében szerepel a „perl” szó.¹⁸⁸

A Perl egyik alternatívája a **Ruby**, mely egy értelmezett script-elő objektum-orientált nyelv. A Perl-hez hasonlóan rengeteg csomagra van bontva a Debian-ban. Illesztőt találhatunk nagyon sok más nyelvhez és persze az Apache-nak is van ilyen modulja. Ekkor a Ruby CGI script-ek natív módban lesznek értelmezve, ezzel gyorsítva a végrehajtást. Az `eruby` csomaggal Ruby nyelvű elemeket szűrhatunk HTML fájlalba. A Ruby-val még „csak” 32 csomag foglalkozik.

Nézelődhetünk még a **Python** és **Pike** nyelvek területén is, ők is alkalmasak ilyen feladatok ellátására. Fellelhető a Perl-ről¹⁸⁹ és a CGI-ről¹⁹⁰ magyar nyelvű leírás is.

3. Alternatívák SQL szerverre

A **MySQL** mellett természetesen sok más lehetőség is akad. A másik legelterjedtebb adatbázis-szerver linux alatt a **PostGreSQL**. Ennek a felhasználási területe nem annyira az Internet - Web, mint inkább az alkalmazói programok adatbázis-háttere. A **PostGres** valamivel lassabb, mint a **MySQL**, (ezért javasolják a **MySQL**-t Web-szerverekhez, ahol inkább a sebesség a súlypont) viszont képes a tranzakciók kezelésére, amire a másik nem. A **PostGres** továbbá rendelkezik egy minimális objektum-orientált kódrésszel is. A **PostGres** is folyamatos fejlesztés alatt áll. Az SQL-92-es szabvány nagy részét implementálták már, de még vannak hiányosságok. Mondhatni, a **PostGres** többet tud, mint a **MySQL**, több funkció van már implementálva. A Debian-ban jelenleg a 6.5.3-as verzió található. A legfrissebb változata a 7.0-s.

¹⁸⁸ `dpkg -l *perl* | wc -l`

¹⁸⁹ <http://www.vmg.sulinet.hu/vmghome/szamtech/perl>

¹⁹⁰ <http://www.vmg.sulinet.hu/vmghome/szamtech/cgi>

Ez a program is több csomagra van szétszedve:

postgresql	Az alapcsomag, ez tartalmazza a szervert
postgresql-client	Karakteres kliens programok az adatbázisok kezeléséhez.
postgresql-contrib	Apró kis bővítmények, hasznos segédfunkciók tárháza, ezek még főleg fejlesztés alatt lévő kódrészek.
postgresql-test	Fejlesztők részére
postgresql-dev	Fejléc fájlok, fejlesztők részére
postgresql-pl	Procedurális programozási nyelv, fejlesztőknek
postgresql-doc	Teljes dokumentáció
odbc-postgresql	ODBC illesztő
www-pgsql	Web-es programozási interfész (külön program!)

11. táblázat - PostgreSQL csomagok a Debian-ban

Természetesen rengeteg programozási nyelvhez is van *PostGres* illesztő, mint pl. a *Python*, *Pike*, *PHP*, stb.

Végeredményben, egy egész jó ingyenes adatbázis-szerver programmal van dolgunk. Ha szükségünk van tranzakció kezelésre, és nem a sebesség számít, akkor választhatjuk ezt a MySQL helyett.

Link: <http://www.postgres.org>

4. Alternatívák a távoli bejelentkezésre

A terminálos bejelentkezésre sok alternatíva létezik. Kezdetben volt az *rsh/rlogin/rcp*¹⁹¹ páros. Mivel ezek kódolatlan csatornát hoznak létre a két gép között ezért csak az Internettől elzárt és lehallgatás-biztos helyeken illene használni, vagy ott se. Ezeket ma már „épesű” ember nem használja távoli bejelentkezésre, hiszen bárhol lehet egy „szaglászó” program, ami csak a mi jelszónkra vár.

A különböző SSH implementációk ezt a héjat (shell-t) egy titkosított csatornán keresztül valósítják meg. Mivel az SSH RSA kódolást használ ezért a többi kriptográfiát tartalmazó programmal együtt csak a non-US szerverekről tölthetőek le.

A másik lehetőség a titkosított *telnet* használata. Ez az (Open)SSLeay könyvtárt használva hoz létre titkos csatornát. Ha nem talál a másik oldalon SSL-képes *telnet* démont, akkor „visszaesik” hagyományos *telnet*-té.

¹⁹¹ Remote Shell, Remote Login, Remote Copy

ssh	1.2.2	Ez a változat az OpenSSH programcsomagot tartalmazza, amely az OpenBSD operációs rendszerről került át ide. Igazi előnye a licensz.
ssh-askpass	0.99	Az X grafikus felület alatt megkérdezi a jelszót.
ssh-askpass-gnome	1.2.2	ua., de <i>gnome-os</i> változat.
ssh-askpass-ptk	1.2.2	ua., de <i>perl-tk-s</i> változat
ssh-nonfree	1.2.27	Az SSH eredeti, de nem szabad megvalósítása
ssh-askpass-nonfree	1.2.27	ua., de nem szabad változat
ssh-socks	1.2.27	SSH SOCKS támogatással
ssh2	2.0.13	Az SSH eredeti második generációs változata
telnet-ssl	0.16.1	SSL-képes <i>telnet</i> kliens
telnetd-ssl		SSL-képes <i>telnet</i> démon

12. táblázat - Alternatív csomagok távoli bejelentkezésre a Debian-ban

5. Alternatívák az egyéb programokra

A Debian-ban rengeteg olyan eset fordul elő, hogy egy azonos feladatra több program nyújthat megoldást. Pl. rengeteg héjprogram közül választhatunk igényeink szerint.

ash	A NetBSD shell
bash	A GNU szabványos shell-je
csch	*BSD C shell
es	Az rc shell bővítve
esh	Lisp szintaxisú shell
kiss	Bash-stílusú, sok beépített paranccsal
lsh	*DOS szintaxis
rc	Az AT&T Plan 9 shell implementációja
sash	Statikusan linkelt shell
tcsh	TENEX C Shell, (BSD <i>csch</i> alapján)
zsh	Sokfunkciós shell (<i>ksh</i> alapján)

13. táblázat - Shell-ek a Debian-ban

Amint a fenti táblázatban látható, mindenki megtalálhatja a más rendszerekben már jól megszokott héját.

A fontos az, hogy szánjunk rá időt és böngésszük végig a csomaglistát. Keressük meg az azonos funkciókat kitöltő csomagokat és válasszuk ki a nekünk szimpatikusakat. Ha nem tudjuk melyik a jobb, próbáljuk meg mindet és később döntsünk, melyik marad.

VIII. Összegzés

Végére értünk a munkának, gondolhatja a kedves olvasó. Sajnos ezzel korántsem. Az egész témakör olyan szerteágazó és olyan gyorsan változó, hogy ha lépést akarunk tartani a fejlődéssel, kénytelenek vagyunk folyamatosan tanulni, tesztelni, fejleszteni, alkalmazni a legújabb szoftvereket és hardvereket.

Mint ahogy már az Előszóban is említettem, kétséges, hogy ez az egész exponenciális fejlődési folyamat igazából a hasznunkat szolgálja-e, de ezt pár év távlatából nem ítéldhetjük meg.

Mindenesetre, az informatika használatának ugrásszerű növekedése, a kereskedelem Internettel való felgyorsítása és „kényelmesebbé” tétele megkívánta, szinte életre hívta a szabad szoftver mozgalmakat, mivel egyszerűen olyan sok új kiszolgáltót és munkaállomást kell munkába állítani, melyeknek kommerciális szoftverekkel való ellátása anyagi okok miatt egyenesen hátráltatná a fejlődést.

A jövő az Interneté – mondják. Az Internet a UN*X alapú rendszerekből nőtt ki magát. A szabad szoftverek pedig az Internetből „nőttek ki”. Evidens tehát, hogy legtesthezálóbb felhasználási területük maga az Internet.

Természetesen az új technológiákat fel is kell tudni használnunk, ehhez pedig (ön)képzés, oktatás szükséges. Remélem ezzel az írással sokaknak sokat segíték a Debian GNU/Linux 2.2 rendszer telepítésének és behangolásának elsajátításában. Továbbá fontos kiemelni, hogy ez a rendszer csupán egy (talán az egyik legjobb) a sok jó szabad szoftverre épülő rendszer közül. Akik az igazi UN*X világból érkeznek, talán a *BSD variánsokat jobbnak találják. Akik a felhasználói oldalról érkeznek, egyes kommerciális Linux terjesztést sokkal könnyebben kezelhetőnek fogják találni. Végeredményben a legtöbb itt bemutatott program megtalálható a többi szabad szoftveres operációs rendszerben is, ezért nem csak a Debian-osok vehetik hasznát ennek az írásnak.

Az eszköz szabadon megválasztható. A cél az, hogy a szabad és ingyenes szoftverek minél szélesebb körben elterjedjenek, ezzel ellensúlyozva az Interneten uralkodni próbáló óriáscégeket, továbbá lehetővé tenni mindenkinek, hogy kevés befektetéssel részese legyen az Online kereskedelemnek, az E-Bussines-nek, az Internetnek.

Legyen az magánszemély, oktatási, állami intézmény, vagy profit-orientált cég, mindenki jól jár a szabad szoftverekkel, hiszen „olcsó” hardver segítségével és ingyenes szoftverekkel saját maga által alakítható belépőt kap az új világba.

Irodalomjegyzék

- (1) Olaf Kirch: Linux hálózati adminisztrátorok kézikönyve, Kossuth kiadó, 1998.
- (2) Garzó András: Nem paranoia! in Chip Magazin, 1999. Augusztus - 8. sz., Vogel kiadó, p. 123-124
- (3) David Medinets: PHP3 Programming Browser-based Applications, McGraw-Hill, NY, 2000
- (4) Knapp Gábor: Operációs rendszerek, Budapest, LSI Oktatóközpont, 1998.
- (5) Bakos Tamás – Zsadányi Pál: Operációs rendszerek, Budapest, LSI Oktatóközpont, 1989.
- (6) Markó Imre: PC-k konfigurálása és installálása, LSI Oktatóközpont, Bp. 1999.
- (7) Czákó Krisztián: A 2.2-es Kernel beállítása, in CHIPTÁR 14., Vogel Publishing Kft., Budapest, 1998. p 40-65
- (8) Paul DuBois: MySQL, New Riders, 1st Edition December 1999
- (9) Judith S. Bowman, Sandra L. Emerson and Marcy Darnovsky: The Practical SQL Handbook: Using Structured Query Language, Second Edition, Addison-Wesley, <http://www.awl.com>, H.n., É.n.
- (10) Martin Gruber: Understanding SQL, Publisher Sybex 510 523 8233, Alameda, CA USA, É.n.
- (11) Dr. Szelezsán János: Adatbázisok, LSI Oktatóközpont, Budapest, É.n.
- (12) Pulai-Sziklássy-Tóth-Udvaros: Védd magad az interneten, Kossuth Kiadó, 1997
- (13) Szabó Péter: PHP3 vs. CGI, in Chip Magazin 1999. október – 10. sz., Vogel kiadó, Budapest, p. 185-187.
- (14) Stefan Strobel – Volker Elling: Linux, Kossuth kiadó, 2000.
- (15) Jedlovsky Pál: Unix lépésről-lépésre, LSI Oktatóközpont, É.n.
- (16) Chapman, D.B.-Zwicky, E.D.: Building Internet Firewalls, O'Really & Associates, 1995.
- (17) Cheswic, W.R.-Belowin, S.M.: Firewalls and Internet Security: Repelling the Willy Hacker, Addison-Wesley, 1994.
- (18) Hare C.-Siyan K.: Internet Firewalls and Network Security, 2/E, New Riders, 1996.
- (19) Czákó Krisztián: Az Apache webszerver, in CHIPTÁR: Linux 2.2., Vogel Kiadó, Budapest, 1999.
- (20) Revuen M. Lerner: Protecting your site with access controls, in Linux Journal, May 1998, SSC. Inc., p. 84-89
- (21) Eddie Harari: Post-Installation Security Procedures, in Linux Journal, December 1999, SSC. Inc., p. 76-79.
- (22) Czákó Krisztián: IP-láncok, in Chip Magazin 1999. Április – 4. szám, Vogel kiadó, Budapest, p. 154-156.
- (23) Czákó Krisztián: IP-láncok, in Chip Magazin 1999. Május – 5. szám, Vogel kiadó, Budapest, p. 188-189.
- (24) Paul Russel: Security Basics, in Linux Magazine, November 1999. p. 56,57,76
- (25) Paul Russel: Keeping the TCP/IP stream flowing, in Linux Magazine, August 1999. p. 54-57.
- (26) S. Garfinekl & G. Spafford: Web security and commerce, O'Reilly & Associates, 1997., H.n.
- (27) A. Ghosh: E-commerce security, John Wiley & Sons, 1998. H.n.
- (28) Nalneesh Gaur: Accessing the security of your web applications, in Linux Journal, April 2000, p. 74-78
- (29) Lajbér Zoltán: SaMBa - kapocs a rendszerek között. CHIPTár 9. Budapest, Vogel Publishing Kft, 1998.
- (30) Richard Petersen: Linux referenciakönyv – könnyen is lehet, Panem-McGraw-Hill, Budapest, 1997
- (31) Tóth J. Szabolcs: PC vírusok, LSI Oktatóközpont, 1999.
- (32) Moray Gábor: Beszélgetés Linus Torvaldsszal, a Linux megalkotójával. PC World, 1998. December

Függelék

1. A GPL v2 licenz magyar nyelvű fordítása

GNU GPL Magyar fordítás

A 2. Verzió (1991. június) fordítása

Copyright 1989, 1991 Free Software Foundation, Inc. 675 Mass Ave, Cambridge, MA 02139, USA.

Bárki terjesztheti, másolhatja a dokumentumot, de a módosítása nem megengedett. A fordítás csak tájékoztató jellegű és jogi szempontból csakis az angol eredeti a mérvadó.

Előszó

A legtöbb program felhasználói jogosultságai azzal a szándékkal készültek, hogy minél kevesebb lehetőséget adjanak a terjesztéshez illetve a szoftver megváltoztatásához. A GNU GPL ezzel szemben minél több jogot kíván biztosítani a szabad szoftverek terjesztéséhez és módosításához, hogy a szoftver ingyenes lehessen az összes felhasználója számára. Az Általános Közreadási Feltételek szabályai vonatkoznak a Szabad Szoftver Alapítvány legtöbb szoftverére, illetve minden olyan programra, melynek szerzője úgy dönt, hogy ezt használja a szerzői jog megjelölésekor. (A szabad szoftver alapítvány egyes szoftvereire a GNU LGPL vonatkozik - Library GPL.) Bárki használhatja a programjaiban a GPL-t a szerzői jogi megjegyzésnél.

A szabad szoftver megjelölés nem jelenti azt, hogy a szoftvernek nem lehet ára. A GPL dokumentumok úgy készültek, hogy biztosítsák a szabad szoftverek terjeszthetőségét (pénzért ha úgy tetszik), illetve a forráskód hozzáférhetőséget, hogy bárki szabadon módosíthassa azt, ha akarja mindenképpen tudjon erről a lehetőségről.

A szerző jogainak védelmében korlátozásokat kell hozni, melyek megtiltják, hogy bárki megtagadja e jogokat másoktól, vagy ezekről való lemondásra kényszerítene valakit. Ezek a korlátozások bizonyos kötelezettségeket jelentenek azok számára akik a terjesztik vagy módosítják a szoftvert.

Ha valaki ilyen programot terjeszt, akár ingyen akár egy bizonyos összeg fejében, a szoftverre vonatkozó minden jogát tovább kell adnia az ügyfeleinek. Biztosítani kell továbbá, hogy mindenki megkapja, vagy lehetősége legyen hozzájutni a forráskódhoz. Ezen kívül el kell juttatni ezen szabályokat tartalmazó dokumentumot is, hogy mindenki értesülhessen a jogairól.

A jogvédelem két eszköze: (1) a szoftver szerzői jogainak biztosítása (2) ezen szabályok alapján jogok biztosítása a másoláshoz, terjesztéshez és/vagy a szoftver módosításához.

Az egyes szerzők és a magunk védelmében biztosítani akarjuk, hogy mindenki megértse: nincs garancia az ilyen szoftverekre. Ha a szoftvert mások módosították és továbbterjesztették, mindenkinek, aki a módosított változatot kapja, tudnia kell, hogy az nem eredeti. Így a mások által okozott hibáknak nem lehet semmiféle hatása az eredeti szerző hírnevére.

Végül, mivel a szabad szoftver létét fenyegetik a szoftverszabadalmak, el szeretnénk kerülni annak veszélyét, hogy a szabad szoftver terjesztői szabadalmat jegyezthessenek be a szoftverre, így tulajdonukká téve azt. Ennek megelőzéséhez tisztázni kívánjuk: szabadalom szabad szoftverrel kapcsolatban csak mindenki által szabad használatra jegyezhető be, vagy egyáltalán nem jegyezhető be.

A pontos szabályok és a másolásra, terjesztésre, módosításra vonatkozó feltételek következnek.

A másolásra, terjesztésre és módosításra vonatkozó feltételek és szabályok

0. Ezek a jogok vonatkoznak bármely olyan programra vagy munkára, melynek a szerzői jogi megjegyzésében a jog tulajdonosa a következő szöveget helyezte el: Terjeszthető a GNU GPL-ben foglaltak alapján. A következőkben a „Program” megjelölés vonatkozik bármely programra, vagy munkára, a „Programon alapuló munka” pedig magát a Programot, vagy a Programot felhasználó szerzői joggal védett munkát jelenti, vagyis olyan munkát, mely tartalmazza a Programot, vagy annak egy részletét, módosítottan vagy módosítatlanul és/vagy más nyelvre fordítva. (Ezentúl a fordítás minden egyéb megkötés nélkül beletartozik a „módosítás” fogalmába.)

A másoláson, terjesztésen és módosításon kívül más tevékenységgel nem foglalkozik ez a dokumentum, azokat hatályon kívülnek tekinti. A Program futtatása nincs korlátozva, illetve a Program kimenetére is csak abban az esetben vonatkozik ez a szabályozás, ha az tartalmazza a Programon alapuló munka egy részletét (attól függetlenül, hogy ez a Program futtatásával jött-e létre). Ez tehát a Program működésétől függ.

1. A Program forráskódja másolható és terjeszthető módosítás nélkül bármely adathordozón, feltéve, hogy minden egyes példányon pontosan szerepel a megfelelő szerzői jogi megjegyzés, illetve a garanciavállalás elutasítása. Érintetlenül kell hagyni minden erre a szabályozásra és a garancia teljes hiányára utaló szöveget, és ezt a dokumentumot is el kell juttatni mindazokhoz, akik a Programot kapják.

Kérhető pénz az adatok fizikai továbbítása fejében, illetve díjazás fejében lehet garanciás támogatást adni a Programhoz.

2. A Program, vagy egy darabja módosítható, mely így az egy a Programon alapuló munkát alkot, a módosítás ezután tovább terjeszthető az 1. részben adott feltételek szerint, ha az alábbi feltételek is teljesülnek:

a. A módosított fájlokat el kell látni olyan megjegyzéssel, mely feltünteti a módosítást végző nevét és a módosítások dátumát.

b. Minden olyan munkát vagy programot, mely részben vagy egészben tartalmazza a Programot vagy a Programon alapul, olyan szabályokkal kell kiadni, hogy annak használati joga harmadik személy részére ingyenesen hozzáférhető legyen, ezen dokumentumban található szabályok alapján.

c. Ha a módosított Program interaktív bemenetet használ, akkor azt úgy kell elkészíteni, hogy a megszokott módon történő indításkor megjelenítsen egy üzenetet a megfelelő szerzői jogi megjegyzéssel és a garancia hiányára utaló közléssel (vagy éppen azzal az információval, hogy minként juthat valaki garanciához), illetve azzal az információval, hogy bárki terjesztheti a Programot eme feltételek alapján. Ezen kívül utalást kell tenni rá, hogy miként olvashatja el a felhasználó ezt a dokumentumot. (Kivétel: ha a Program interaktív ugyan, de nem jelenít meg hasonló üzenetet, akkor a Programon alapuló munkának sem kell ezt tennie.)

Ezek a feltételek a módosított munkára, mint egészre vonatkoznak. Ha a munka egy azonosítható részei nem a Programon alapulnak, függetlenül elkülönülten azonosíthatók, akkor ez a szabályozás nem vonatkozik ezekre a részekre, ha azok külön munkaként vannak terjesztve. Viszont, ha ugyanez a rész az egész részeként kerül terjesztésre, és az egész a Programon alapuló munka, akkor az egész terjesztése csak ezen dokumentum alapján lehetséges, mely ebben az esetben a jogokat minden egyes felhasználó számára kiterjeszti az egészre tekintet nélkül arra, hogy mely részt ki írta. Ezen szövegrésznek nem az a célja, hogy a mások jogait elvegye vagy korlátozza a kizárólag saját maga által írt munkákra. A cél, hogy a jogok gyakorlása szabályozva legyen a Programon alapuló, illetve a gyűjteményes munkák terjesztése esetében is.

Ezen kívül más munkák, melyek nem a Programon alapulnak, a Programmal (vagy a Programon alapuló munkával) közös adathordozón vagy adattárolón szereplése nem jelenti ezen szabályok érvényességét azokra is.

3. A program (vagy a programon alapuló munka a 2. szakasz alapján) másolható és terjeszthető tárgykódú vagy végrehajtható kódú formájában az 1. és 2. szakaszban foglaltak szerint, amennyiben az alábbi feltételek is teljesülnek:

a. A teljes, gép által értelmezhető forráskód kíséri az anyagot, melynek terjesztése az 1. és 2. szakaszban foglaltak szerint történik, szoftverterjesztésre használt hordozón; vagy

b. Egy legalább három évre szóló írásos ajánlat kíséri az anyagot, mely szerint bármely külső személynek rendelkezésre áll a teljes gép által értelmezhető forráskód, a fizikai továbbítást fedező összegnél nem nagyobb díjért az 1. és 2. szakaszban foglaltak szerint szoftverterjesztésre használt adathordozón; vagy

c. Olyan tájékoztatás kíséri az anyagot, mely tartalmazza az írásos ajánlat szövegét a forráskód biztosítására. (Ez az alternatíva csak nem kereskedelmi terjesztés esetén alkalmazható, abban az esetben, ha a terjesztő a Programhoz a tárgykódú vagy forráskódú formájában jutott hozzá az ajánlattal együtt a b. cikkelynek megfelelően.)

Egy munka forráskódja a munkának azt a formáját jelenti, melyben a módosításokat szokás végezni. Egy végrehajtható program esetében a teljes forráskód jelenti a modulok forráskódját, a kapcsolódó felületkezelő definíciós fájlokat, és a fordítást vezérlő parancsfájlokat. Egy speciális kivételként a forráskódnak nem kell tartalmaznia az operációs rendszer főbb részeit (kernel fordítóprogram stb.), melyen a végrehajtható kód fut, hacsak nem tartozik ehhez maga a program is.

Ha a végrehajtható program vagy tárgykód terjesztése a forráskód hozzáférését egy megadott helyen biztosító ajánlattal történik, ez az ajánlat egyenértékű a forráskód terjesztésével, még akkor is, ha másoknak így nem kell a forrást lemásolniuk a tárgykóddal együtt.

4. A Programot csak ebben a dokumentumban leírtaknak megfelelően lehet lemásolni, terjesztetni, módosítani, rá jogokat bejegyezni. Az egyéb módon való másolás, módosítás, terjesztés, jogok bejegyzése semmisé teszi a dokumentumban közzétett jogosultságokat. Akik azonban a jogaikat ennek a szerzői jogi szabályozás keretei között kapták, azok joga mindaddig megmarad, amíg teljesen megfelelnek a leírtaknak.

5. Nem kell elfogadni ezt a szabályozást, mivel aláírni sem kell. Ezen kívül viszont semmi más nem adhat jogokat a Program továbbterjesztésére és módosítására. Ezeket a cselekedeteket a törvény bünteti, ha nem ennek a szerzői jogi szabályozásnak a keretei között történnek. Mindezek miatt a Program (vagy a Programon alapuló munka) terjesztése vagy módosítása ezen dokumentum szabályinak elfogadását jelenti.

6. Minden alkalommal, amikor a Program (vagy azon alapuló munka) továbbadása történik, a Program „vevője” automatikusan hozzájut a Program eredeti tulajdonosának szerzői jogait tartalmazó dokumentumhoz, mely biztosítja a Program másolását és terjesztését eme szabályok szerint. Nem lehet semmi módon további korlátozásokat hozni a „vevő” számára ezen szabályok betartása céljából. Más szavakkal: a Program továbbadója nem felelős más személyekkel betartatni ezeket a szabályokat.

7. Ha bírósági határozat vagy más szabadalmi kötöttségek miatt olyan feltételek állnak elő, melyek ellentétesek e szabályozással, ezek nem mentik fel a terjesztőt a feltételek figyelembevétele alól. Ha a terjesztés nem lehetséges ezen szabályozás szerint, akkor egyáltalán nem lehetséges. Például, ha egy szabadalmi szerződés nem engedi meg egy program tiszteletdíj nélküli terjesztését, akkor az egyetlen módja, hogy eleget tegyen valaki mindkét szabályozásnak az, hogy eláll a továbbfejlesztett program terjesztésétől.

Ha ennek a szakasznak bármely része nem érvényesül, vagy nem érvényesíthető valamely körülmény folytán, akkor a szakaszt kell mérlegelni, egyéb esetekben a szakasz, mint egész alkalmazandó.

Ennek a szakasznak nem az a célja, hogy a szabadalmak vagy egyéb hasonló jogok elutasítására bírjon bárkit is. Mindössze meg szeretné védeni a szabad szoftver terjesztés rendszerének egységét, melyet a szabad közreadást szabályozó feltételrendszerek teremtenek meg. Sok ember nagylelkű közreműködése folytán igen nagyszámú és változatos szoftver terjesztése történik ezen a módon, mely nagyban függ ennek a feltétel-rendszernek állandó betartásán. Minden esetben a szerző/adományozó dönti el, hogy művét mely rendszer szerint teszi közzé. Ezt a döntést a jogok felhasználója nem befolyásolhatja.

Ez a szakasz pontosan szeretné tisztázni a szabályozás hátralevő részének lehetséges következményeit.

Ha a Program terjesztése és/vagy használata egyes országokban nem lehetséges szabadalmak vagy szerzői jogokkal védett kapcsolódási felületek miatt, akkor a Program szerzői jogainak eredeti tulajdonosa, aki a Programot ezen szabályozás alapján adja közre, egy földrajzi megkötést adhat a terjesztésre, és egyes országokat kizárhat. Ekkor a terjesztés csak azokban az országokban lehetséges, amelyek nem lettek ilyen módon kizárva. Ebben az esetben ennek a szabályozásnak kell tartalmazni az ilyen megkötéseket is is.

9. A Szabad Szoftver Alapítvány időnként a dokumentum felülvizsgált illetve újabb változatait adja ki. Ezek az újabb dokumentumok az előzőek szellemében készülnek, de részletekben különböznek, hogy új problémákat kezelhessenek.

A dokumentum minden változata egy meghatározott verziószámmal ellátva jelenik meg. Ha a program szerzői jogi megjegyzésében egy bizonyos vagy annál újabb verzió van megjelölve, akkor lehetőség van akár a megjelölt, vagy a Szabad Szoftver Alapítvány által kiadott későbbi verzióban leírt feltételek követésére. Ha nincs ilyen megjelölt verzió, akkor a Szabad Szoftver Alapítvány által valaha kibocsátott bármelyik dokumentum alkalmazására lehetőség van.

10. A Programot más szabad szoftverbe, melynek szerzői jogi szabályozása különbözik a GPL-től, akkor lehet beépíteni, ha a szerzőtől erre engedélyt lehet szerezni. Abban az esetben, ha a program szerzői jogainak tulajdonosa a Szabad Szoftver Alapítvány, akkor a Szabad Szoftver Alapítvány címére kell írni. Az alapítvány egyes esetekben ezt engedélyezi. A döntés a következő két cél szem előtt tartásával fog megtörténni: Megmaradjon a Programon alapuló munkák szabad státusa; Valamint segítse elő a szoftver újrafelhasználását és megosztását.

NINCS GARANCIÁVÁLLALÁS

11. MIVEL A PROGRAM HASZNÁLATI JOGA DÍJMENTES, A PROGRAMHOZ NEM JÁR GARANCIA AZ IDEVONATKOZÓ JOGSZABÁLYNAK MEGFELELŐEN. AMENNYIBEN A SZERZŐI JOGOK TULAJDONOSAI ÍRÁSBAN MÁSKÉNT NEM NYILATKOZNAK, A PROGRAM "ÚGY AHOGY VAN" KERÜL KIADÁSRA MINDENFÉLE GARANCIÁVÁLLALÁS NÉLKÜL.

A PROGRAMMAL KAPCSOLATBAN NINCS SEM SZÁRMAZTATOTT, SEM EGYÉB GARANCIÁVÁLLALÁS BELEÉRTVE DE NEM KIZÁRÓLAGOSAN A FORGALOMBAHOZHATÓSÁGRA VAGY ALKALMAZHATÓSÁGRA VONATKOZÓ GARANCIÁKAT. A PROGRAM MINŐSÉGÉBŐL ÉS MŰKÖDÉSÉBŐL FAKADÓ ÖSSZES KOCKÁZAT A FELHASZNÁLÓT TERHELI. HA A PROGRAM HIBÁSAN MŰKÖDIK, A FELHASZNÁLÓNAK MAGÁNAK KELL VÁLLALNIA A JAVÍTÁSHOZ SZÜKSÉGES MINDEN KÖLTSÉGET.

12. AMENNYIBEN A HATÁLYOS JOGSZABÁLYOK, VAGY A SZERZŐI JOGOK TULAJDONOSAI ÍRÁSBAN MEGÁLLAPODÁSBAN MÁSKÉNT NEM RENDELKEZNEK, SEM A PROGRAM SZERZŐJE SEM MÁSOK, AKIK MÓDOSÍTOTTÁK ÉS/VAGY TERJESZTETTÉK A PROGRAMOT A FENTIEKNEK MEGFELELŐEN, NEM TEHETŐK FELELŐSSÉ KÁROKÉRT MELYEK LEHETNEK VÉLETLENEK, VAGY MEGHATÁROZOTT KÖRÜLMÉNYEK MIATT TÖRTÉNEK (BELEÉRTVE DE NEM KIZÁRÓLAGOSAN A AZ ADATVESZTÉST ÉS A HELYTELEN ADATFELDOLGOZÁST, VALAMINT A MÁS PROGRAMOKKAL VALÓ HIBÁS EGYÜTTMŰKÖDÉST), AKKOR SEM, HA EZEN FELEK TUDATÁBAN VOLTAK ILYEN KÁROK KELETKEZÉSÉNEK LEHETŐSÉGÉNEK.

FELTÉTELEK ÉS SZABÁLYOK VÉGE

Függelék: Hogyan alkalmazhatóak ezek a szabályok egy új programra

Ha valaki egy új programot készít és szeretné hogy az a többi ember számára a lehető leginkább hasznos legyen, annak az a legjobb módja,

hogy szabad szoftverré teszi azt, megengedve bárki számára a szabad másolást és módosítást ezen szabályok alapján.

Ehhez a következő megjegyzést kell csatolni a programhoz. A legbiztosabb ezt minden egyes forrásfájl elejére beírni, így közölve leghatásosabban a garancia visszautasítását. Minden fájl ezen kívül kell, hogy tartalmazzon egy „copyright” sort és egy utalást arra helyre, ahol a teljes szöveg található.

Egy sor mely megadja a program nevét, és leírja, hogy mit csinál. Copyright 19yy a szerző neve Ez egy szabad szoftver; terjeszthető illetve módosítható a GNU Általános Közreadási Feltételek dokumentumában leírtak szerint -- 2. vagy későbbi verzió --, melyet a Szabad Szoftver Alapítvány ad ki.

Ez a program abban a reményben kerül közreadásra, hogy hasznos lesz, de minden egyéb GARANCIA NÉLKÜL, az eladhatóságra vagy valamely célra való alkalmazhatóságra való származtatott garanciát is beleértve. További részletekért lásd a GNU Általános Közreadási Feltételek dokumentumát.

A programmal együtt kellett, hogy érkezzen egy példány a GNU Általános Közreadási Feltételek dokumentumából is. Ha mégsem akkor ezt a Szabad Szoftver Alapítványnak küldött levélben jelezni kell. A szabad szoftver alapítvány címe: Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

A programhoz csatolni kell azt is, hogy miként lehet kapcsolatba lépni a szerzővel, elektronikus vagy hagyományos levél küldésével.

Ha a program interaktív, a következőhöz hasonló üzenettel lehet ezt megtenni a program indulásakor:

Gnomovision version 69, Copyright (C) 19yy a szerző neve A Gnomovision programhoz SEMMIFÉLE GARANCIA NEM JÁR; A részletes tájékoztatáshoz ezt kell begépelni: „show w”. Ez egy szabad szoftver; bizonyos feltételek mellett terjeszthető illetve módosítható. A részletes tájékoztatáshoz ezt kell begépelni: `show c'.

A „show w” és „show c” képzeletbeli parancsok, és a GNU Általános Közreadási Feltételek megfelelő szakaszát kell megjeleníteniük. Természetesen a valódi parancsok lehetnek a „show w” és a „show c”-től különbözőek is, lehetnek akár egérkattintások vagy menüpontok is a programnak megfelelően.

Ha szükséges, meg kell szerezni a munkáltatótól (programozó esetében) vagy iskolától a program szerzői jogairól való lemondás igazolását.

Erre itt egy példa:

Az Adócsaló BT ezennel lemond minden szerzői jogi érdekelttségéről a „Gnomovision” programmal kapcsolatban, melyet

Hekker Jani írt.

Aláírás: Maffy Jocó, 1987. április 1.

Maffy Jocó, alelnök

A GPL általános közreadási feltételek dokumentuma nem engedi meg, hogy szabad szoftver része legyen szabadalommal védett programnak. Ha a program egy eljáráskönyvtár akkor inkább a más programokkal való összefűzését célszerű megengedni. Ha ez a cél akkor a GNU LGPL dokumentumot lehet alkalmazni, mely ilyen eljáráskönyvtárak közreadását szabályozza.

2. A BSD licenz

Copyright (c) The Regents of the University of California.

All rights reserved.

A forrás- és bináris formában történő terjesztés, módosítással, vagy anélkül akkor engedélyezett, ha a következő feltételek teljesülnek:

1. A forráskód terjesztésekor meg kell őrizni a fenti szerzői jogi megjegyzést, ezt a feltétellistát és a következő nyilatkozatot.
2. Bináris formában történő terjesztéskor reprodukálni kell a fenti szerzői jogi megjegyzést, ezt a feltétellistát, a következő nyilatkozatot a dokumentációban, valamint a csomaggal biztosított egyéb anyagokat.
3. Ennek a szoftvernek a szolgáltatásait vagy használatát említő összes hirdetési anyag a következő közönetnyilvánítást kell, hogy tartalmazza:
Ez a termék a University of California, Berkeley és külső munkatársai által fejlesztett szoftver.
4. Sem az egyetem neve, sem pedig a külső munkatársainak neve előzetes írásbeli engedély nélkül nem használható fel a szoftverből származtatott termékek hitelesítésére, vagy reklámozására.

EZT A SZOFTVERT AZ EGYETEM IGAZGATÓTANÁCSÁNAK A TAGJAI ÉS A KÜLSŐ MUNKATÁRSÁK ÚGY BIZTOSÍTJÁK, „AHOGY VAN”, ÉS SEMMILYEN NYÍLT VAGY BURKOLT GARANCIA – BELEÉRTVE, DE NEM ERRE KORLÁTOZVA AZ ELADHATÓSÁGOT VAGY EGY ADOTT CÉLRA VALÓ ALKALMATOSSÁGOT – NEM ÉRVÉNYESÍTHETŐ. AZ EGYETEM IGAZGATÓTANÁCSÁNAK TAGJAI ÉS A KÜLSŐ MUNKATÁRSÁK NEM

VONHATÓK FELELŐSSÉGRE A SZOFTVER HASZNÁLATÁBÓL EREDŐ SEMMILYEN KÖZVETLEN, KÖZVETETT, VÉLETLENSZERŰ, KÜLÖNLEGES, PÉLDAADÓ VAGY SZÜKSÉGSZERŰ KÁROKÉRT (BELEÉRTVE, DE NEM ERRE KORLÁTOZVA A HELYETTESÍTŐ TERMÉKEK VAGY SZOLGÁLTATÁSOK BESZERZÉSÉT, ÜZEMKIESÉST, ADATVESZTÉST, ELMARADT HASZNOT VAGY ÜZLETMENET MEGSZAKADÁSÁT), BÁRHOGY IS KÖVETKEZETT BE, VALAMINT A FELELŐSSÉG BÁRMILYEN ELMÉLETÉVEL – AKÁR SZERZŐDÉSBEN, AKÁR OKOZOTT KÁRBAN (BELEÉRTVE A HANYAGSÁGOT ÉS EGYEBET), AKKOR IS, HA AZ ILYEN KÁR LEHETŐSÉGÉRE FELHÍVTÁK A FIGYELMET.

3. A Debian „Social Contract” (Társadalmi szerződés) / DFSG

A Social Contract

Debian, the producers of the Debian GNU/Linux system, have created the Debian Social Contract. The contract, initially designed as a set of commitments that we agree to abide by, has been adopted by the free software community as the basis of the Open Source Definition.

"Social Contract" with the Free Software Community

1. Debian Will Remain 100% Free Software

We promise to keep the Debian GNU/Linux Distribution entirely free software. As there are many definitions of free software, we include the guidelines we use to determine if software is "free" below. We will support our users who develop and run non-free software on Debian, but we will never make the system depend on an item of non-free software.

2. We Will Give Back to the Free Software Community

When we write new components of the Debian system, we will license them as free software. We will make the best system we can, so that free software will be widely distributed and used. We will feed back bug-fixes, improvements, user requests, etc. to the "upstream" authors of software included in our system.

3. We Won't Hide Problems

We will keep our entire bug-report database open for public view at all times. Reports that users file on-line will immediately become visible to others.

4. Our Priorities are Our Users and Free Software

We will be guided by the needs of our users and the free-software community. We will place their interests first in our priorities. We will support the needs of our users for operation in many different kinds of computing environment. We won't object to commercial software that is intended to run on Debian systems, and we'll allow others to create value-added distributions containing both Debian and commercial software, without any fee from us. To support these goals, we will provide an integrated system of high-quality, 100% free software, with no legal restrictions that would prevent these kinds of use.

5. Programs That Don't Meet Our Free-Software Standards

We acknowledge that some of our users require the use of programs that don't conform to the Debian Free Software Guidelines. We have created "contrib" and "non-free" areas in our FTP archive for this software. The software in these directories is not part of the Debian system, although it has been configured for use with Debian. We encourage CD manufacturers to read the licenses of software packages in these directories and determine if they can distribute that software on their CDs. Thus, although non-free software isn't a part of Debian, we support its use, and we provide infrastructure (such as our bug-tracking system and mailing lists) for non-free software packages.

The Debian Free Software Guidelines (DFSG)

1.Free Redistribution

The license of a Debian component may not restrict any party from selling or giving away the software as a component of an aggregate software distribution containing programs from several different sources. The license may not require a royalty or other fee for such sale.

2.Source Code

The program must include source code, and must allow distribution in source code as well as compiled form.

3.Derived Works

The license must allow modifications and derived works, and must allow them to be distributed under the same terms as the license of the original software.

4.Integrity of The Author's Source Code

The license may restrict source-code from being distributed in modified form only if the license allows the distribution of "patch files" with the source code for the purpose of modifying the program at build time. The license must explicitly permit distribution of software built from modified source code. The license may require derived works to carry a different name or version number from the original software. (This is a compromise. The Debian group encourages all authors not to restrict any files, source or binary, from being modified.)

5.No Discrimination Against Fields of Endeavor

The license must not restrict anyone from making use of the program in a specific field of endeavor. For example, it may not restrict the program from being used in a business, or from being used for genetic research.

7.Distribution of License

The rights attached to the program must apply to all to whom the program is redistributed without the need for execution of an additional license by those parties.

8.License Must Not Be Specific to Debian

The rights attached to the program must not depend on the program's being part of a Debian system. If the program is extracted from Debian and used or distributed without Debian but otherwise within the terms of the program's license, all parties to whom the program is redistributed should have the same rights as those that are granted in conjunction with the Debian system.

9.License Must Not Contaminate Other Software

The license must not place restrictions on other software that is distributed along with the licensed software. For example, the license must not insist that all other programs distributed on the same medium must be free software.

10.Example Licenses

The "GPL", "BSD", and "Artistic" licenses are examples of licenses that we consider "free".

Bruce Perens wrote the first draft of this document and refined it using the comments of the Debian developers during a month-long e-mail conference in June 1997. He later removed the Debian-specific references from the Debian Free Software Guidelines to create "The Open Source Definition". Other organizations may derive from and build on this document. Please give credit to the Debian project if you do.

4. Debian információk

A Debian címe és elérhetősége:

Debian Linux Association

Software in the Public Interest

P.O. Box 70152

Pt. Richmond CA 94807-0512

info@debian.org

5. Rövidítések, szakszavak jegyzéke

account: Felhasználói számla. A számla kifejezés abból ered, hogy a felhasznált processzoridőért, tárolóterületért pénzt számoltak/számolnak fel

API: Application Programming Interface, alkalmazás-programozási felület

APT: Acquisition Package Transfer

BIND: A Berkeley Internet Name Domain kiszolgáló. Egy DNS szerver implementáció.

BIOS: Basic Input / Output System, alapvető kimeneti / bemeneti rendszer, a PC-k Firmware-e

BSD: Berkley Software Distribution

CGI: Common Gateway Interface, Általános Átjáró (kapcsolatteremtő) (programozási) Felület

DMZ: DeMilitarized Zone, semleges zóna

DNS: Domain Name System. Az Interneten gazdagép neveknek IP címekre való leképzésére használt osztott adatbázis.

DoS: Denial of Service, szolgáltatás megbénítása

DVI: DeVice Independent. Platform-független dokumentum-formátum

EGID: (effektív *group id*) - mint *euuid*, csak a csoport-azonosítóra. szuperfelhasználó minden jogokkal rendelkező felhasználó. A felhasználói azonosítója általában 0 szokott lenni minden UN*X-ban, és felhasználói neve (login neve) általában root.

EUID: (effektív user id) - általában egyenlő az uid-del, a felhasználói azonosítóval, de bizonyos esetekben (ún. setuid-bites programoknál) más is lehet. Ilyen módon egy adott folyamatnak több jogot lehet adni, mint ami a folyamat elindítójának van.

FAQ: Frequently Asked Questions, Gyakran Feltett / Ismételt Kérdések (GYIK)

FPU: Floating Point Unit, vagy matematikai társprocesszor.

FSF: Free Software Foundation, Szabad Szoftver Alapítvány

FSN: Free Software Network, Nagy Attila szabad szoftvereket tartalmazó FTP szervere, <ftp.fsn.hu>

FTP: File Transfer Protocol, Az egyik legismertebb fájlátviteli szolgáltatásról elnevezett protokoll.

GID: Csoportazonosító. A UNIX rendszerben minden felhasználó be van osztva egy csoportba. A *gid* annak a csoportnak az azonosítója, amelybe a felhasználó tartozik. (A csoportbeosztás tetszőleges lehet; van olyan rendszer, ahol minden felhasználó egy közös csoportba tartozik.

GMT: Greenwich Mean Time: az egységes csillagászati földi idő, a 0-s időzóna.

GNU: GNU is Not Unix (rekurzív)

GPL: General Public License

HTML: Hyper Text Markup Language, hiper szöveges leíró nyelv

IMAP: Internet Mail Access Protocol: levelezéskor használható levélküldő és fogadó protokoll

Internet: Egy konkrét, a világ minden részére kiterjedő hálózat.

I2O: Intelligent Input/Output: segítségével operációs rendszer független eszközvezérlők írhatóak.

IP: Internet Protocol, Hálózatkezelő protokoll.

ISO: International Standards Organization (nemzetközi szabványügyi szervezet)

LDAP: Lightweight Directory Access Protocol, címtárkezelés

LIDS: Linux Intrusion Detection System Patch, Linux Betörés Detektáló Rendszerfolt

LME: Linux-felhasználók Magyarországi Egyesülete

lo, loopback interfész: olyan virtuális hálózati interfész, mely visszahurkol önmagába

MBR: Master Boot Record, a merevlemez első 512 bájtja.

MLF: Magyar Linux Felhasználók egyesülete

motd: Message of the day, a napi üzenet - a nap üzenete

NCSA: National Center for Supercomputing Applications, University of Illinois

NFS: Network File System, a UN*X klónok hálózati fájlrendszere. Szabványos hálózatkezelési protokoll és szoftvercsomag távoli lemezekben lévő adatok transzparens elérése.

NIS: Network Information System: a Sun cég egy régebbi, nem biztonságos megoldása a felhasználók azonosításra gépek között. A felhasználónak Csak egyszer kell bejelentkeznie a hálózatba, ezután a gépek a NIS segítségével azonosítják azt egymás között. " A NIS egy egyszerű, általános kliens-szerver alapú adatbázis rendszer. Legtöbbször jelszó- és csoportfájlok megosztására használják a hálózaton. Segítséget nyújt a hálózat átlátszóvá tételében azzal, hogy egy bejelentkezéssel használhatjuk az egész hálózat számunkra engedélyezett erőforrásait. A NIS szerver az információkat egyszerű adatbázis-formátumú fájlokban (DBM) tárolja, amelyek lehetővé teszik a gyors keresést. A kliensek RPC hívásokkal tudnak információkat lekérni a szerverről."

PC: Personal Computer, személyi számítógép

PDF: Portable Document Format, szállítható dokumentum formátum

PGRP-ID: Folyamat-csoport azonosítója. Ez egyenlő a folyamat-csoport vezetőjének a pid-jével (minden folyamat tagja valamely folyamat-csoportnak, minden folyamat megalapíthat egy saját folyamat-csoportot, és lehetőség van például egy folyamat-csoport minden tagjának a „kilövésére” egyetlen művelettel).

PID: Folyamat-azonosító.

port, TCP vagy UDP: A portok egy szolgáltatási végpont TCP és UDP absztrakciói. Mielőtt egy folyamat biztosíthat vagy elérhet egy hálózati szolgáltatást, kérnie kell egy portot. A gazdagép IP címeivel együtt a portok egyedileg azonosítják egy TCP kapcsolat két párját.

POSIX: a nemzetközi UNIX szabványosítási hivatal,

PS: PostScript, oldalleíró nyelv

RFC: Request for Common, Internet szabványokat leíró dokumentumok sorozata.

RPC: Remote Procedure Call Távoli gazdagépeken lévő folyamaton belüli eljárások végrehajtására való protokoll.

RTC: Real Time Clock: valós idejű óra a számítógépben.

SGML: Standard General Markup Language, Szabványos Általános Leíró Nyelv

SMP: Symmetric Multi Processing, módszer több processzor használatára egy gépben

SNMP: Simple Network Management Protocol: Hálózati eszközök felügyeletét végzi. Segítségével intelligens hálózati megfigyelő rendszer létesíthető.

SSL: Secure Socket Layer, vagyis Biztonságos Csatorna Réteg.

SQL Structured Query Language, Struktúrált Lekérdező Nyelv

TCO: Total Cost of Ownership, a termék teljes birtoklási ideje alatti költség

TCP: Transmission Control Protocol

TGRP-ID: Terminál group-id. Minden folyamathoz ez is tárolva van. Ez egyenlő annak a folyamatnak a pid-jével, amely a folyamathoz tartozó terminál-(képernyő) fájlt legelőször megnyitotta. Ez általában a legelőször elindult *login shell*.

UDP: User Datagram Protocol

UID: A felhasználó azonosítója (a rendszerben minden egyes felhasználónak egy ilyen egyedi azonosítója van).

UPS: Uninterruptible Power Supply, szünetmentes tápegység

URL: Uniform Resource Locator, egységes erőforrás kijelölő, vagy magyarosabban Web-objektum-cím.

USB: Universal Serial Bus: Új, nagyobb sebességű soros-port szabvány.

VFS: Virtual File System, virtuális fájl-rendszer

Web: háló

XML: eXtensible Markup Language: a HTML-t felváltó, újgenerációs leíró nyelv.

6. Ajánlott RFC-k

<http://www.faqs.org/rfcs/rfc2196.html> címen egy biztonságpolitikai szabályzatot találunk RFC-be foglalva. Továbbá érdemes áttekinteni az 1244 és 1281-es számú RFC-eket is, melyek szintén ezzel a témával foglalkoznak.

Ajánlott biztonsággal kapcsolatos RFC-k:¹⁹²

1108 Security Options for the Internet Protocol

¹⁹² [12 p. 207-209] listájából válogatva

1244 Site Security Handbook
1321 The MD5 Message-Digest Algorithm
1421 Privacy enhancement for Internet electronic mail: Part I: Message encryption and authentication procedures
1422 Privacy enhancement for Internet electronic mail: Part II: Certificate-based key management
1423 Privacy enhancement for Internet electronic mail: Part III: Algorithms, modes and identifiers
1424 Privacy enhancement for Internet electronic mail: Part IV: Key certification and related services
1700 Assigned numbers
1700 On Internet authentication
1948 Defending against sequence number attacks
2069 An extension to HTTP: Digest Access Authentication

7. A LIDS rendszer beállítása

Elérhetőség: <http://www.lids.org> Itt nem tárgyalom a LIDS rendszer működési elvét, ezt mindenki megértheti a dokumentációból. Olvassuk el a csomaghoz adott README és design fájlokat, továbbá töltsük le a Web-oldalról az egyéb dokumentációkat is. A rendszer lényege végül is az, hogy korlátozzuk a root felhasználó jogait, így a behatoló a megfelelő jelszó és hatástalanító módszer nélkül nem tud kárt tenni a rendszerünkben, hiába szerezte meg a root jogosultságokat.

Töltsük le a kernelverziókhöz megfelelő csomagot. Lépjünk be az `/usr/src/linux` könyvtárba. Csomagoljuk ki a fájlt. Foltozzuk meg a kernelt:

```
patch -p1 < lids-0.9.5b/lids-0.9.5b-2.2.16.diff
```

Fordítsuk le a démont: `cd lids-0.9.5b/lidsadm-0.9; make`

Készítsünk el egy jelszót: `lidsadm -P` A kiírt kódot mentjük le, vagy írjuk le, később kelleni fog.

Lépjünk vissza a kernelforrás könyvtárába, majd `make menuconfig` vagy `make xconfig` paranccsal (haladóknak ajánlom a `.config` fájl kézzel való szerkesztését) indítsuk el a kernel konfigurációjának beállítását.

Kernel konfiguráció:

Keressük meg a vége felé lévő *Linux Intrusion Detection System* menüpontot. Itt az azonos nevű első pontot válasszuk „y”-nek. Ekkor megnyílik a lehetőség a többi funkció közötti válogatásban is.

- *Hang up console when raising a security alert:* ezzel lefagyasztaná a konzolt biztonsági riadó esetén. Ezt NE válasszuk ki, mert a konzol biztonságban lesz (remélhetőleg) és távolról fogjuk menedzselni a rendszert.
- *Security alert when execing unprotected programs before sealing LIDS:* Ha a LIDS indítása előtt elindult egy program (mely ez esetben nincs védve), akkor fűjjon riadót. Ezt válasszuk ki, mert ez azt jelentheti, hogy valaki mégis betört és programot telepített a rendszerbe
- *Do not execute unprotected programs before sealing LIDS:* Ne futtassa a nem védett programokat a LIDS indítása előtt. Ezzel vigyáznunk kell, mert ekkor lehet, hogy fel sem áll a rendszerünk, ezért inkább hagyjuk ki.
- *Enable init children lock feature:* Ekkor minden olyan program, mely az init program gyermeke, le lesz védve pl. kilövés ellen. Vagyis ha a behatoló root jogosultságokat is szerzett, mégse tudja leállítani és újraindítani a szolgáltatásokat, démonokat. Ezt jelöljük meg.
- *Try not to flood logs:* Ez az opció vigyáz arra, hogy ha ugyanaz az üzenet kerül nagyon sokszor a naplófájlba, ezzel előbb utóbb tele lehet tölteni a `/var` partíciót. Az opció segítségével az azonos ismétlődő üzenetek kiszűrhetők. Jelöljük meg.
- *Allow switching LIDS protections:* Ekkor menet közben ki és bekapcsolhatjuk a védelmet. Mivel távkarbantartást végzünk, erre szükségünk van. Válasszuk ki. Adjuk meg a mezőbe a kódolt jelszót, növeljük meg a jelszópróbálgatás számát 3-ról, mondjuk 5-re. (Én pl. könnyen félregépelem a jelszavaimat.)
- *Allow remote users to switch LIDS protections:* Ekkor távolról is kikapcsolhatjuk a védelmet, nem csak a konzolról. Válasszuk ki.
- *Allow any program to switch LIDS protection:* Ez veszélyes ne válasszuk ki.
- *Allow reloading config file:* Újraolvashatja a konfigurációs fájlját. Ez hasznos lehet, ha menet közben kell megváltoztatnunk a beállításait. Válasszuk ki.
- *Port scanner detector in kernel:* Kernelszintű portscan érzékelés. Válasszuk ki.

- *Send security alert over network*: A riadó elküldését nem bízta a helyi levelezőre, hanem saját maga képes üzeni. Válasszuk ki.
- *Hide klids kernel thread*: Minden LIDS-hez tartozó programrész elrejtése. A behatoló észre sem veszi, hogy ez a rendszer fut. Válasszuk ki.
- *Remote IP*: annak a gépnek az IP címe (hexa formában), ahová a jelentést küldeni akarjuk. (Adjuk meg pl. annak gépnek az IP címét, amely permanensen elérhető és rendelkezik egy levelező-szerverrel.) Az IP cím egyes decimális tagjait számoljuk át hexadecimális formába, és fordított sorrendbe írjuk be ebbe a mezőbe.
- *Use generic mailer pseudo-script*: Kérjük-e a LIDS -féle levelező script-et, vagy sajáttal rendelkezünk. Válaszunk: Igen. A lenti mezőkbe adjuk meg a forrásgép nevét vagy IP címét., a küldő e-mail címét, a fogadó e-mail címét és a levél témáját. A mi esetünkben ez pl. így nézhet ki: alfa.boresszormegyar.hu, root@alfa.boresszormegyar.hu, rgazda@gyakranolvasom.hu, „Betörés az Alfán!”

Mentsük el az új konfigurációt, fordítsuk le a kernelt (pl. a `make-kpg` paranccsal). Vigyük át a kernel csomagot, a `lidsadm-0.9` könyvtárban lévő `lidsadm (/sbin-be)` és a `lidsadm.1.gz (/usr/local/man/man1-be)` fájlokat a szervergépre. A kernel csomagot telepítsük, majd állítsuk be úgy a `lilo.conf`-ot, hogy egyelőre ne ez legyen a *default* kernel. Másoljuk be a két fájlt a fent megadott könyvtárakba. Most létre kell hoznunk az `/etc/lids.conf` fájlt. Ezt úgy tehetjük meg, hogy a lenti parancsokból készítünk egy *shell-script*-et:

`/root/rules.lids:`

```
lidsadm -A -o /boot -j READ          # ezeket a könyvtárakat ro-ba tesszük
lidsadm -A -o /usr -j READ
lidsadm -A -o /etc -j READ
lidsadm -A -o /lib -j READ
lidsadm -A -o /bin -j READ
lidsadm -A -o /sbin -j READ
lidsadm -A -o /var/log -j APPEND      # ehhez a loghoz csak írni lehet
lidsadm -A -s /usr/sbin/logrotate -o /var/log -j WRITE      # kivéve a logrotate
# Biztosítani kell, hogy az upsd is le tudja állítani a rendszert:
lidsadm -A -s /sbin/shutdown -o UMount -j INHERIT
lidsadm -A -s /sbin/shutdown -o KILL -j INHERIT
lidsadm -A -s /sbin/shutdown -o HD -j INHERIT
```

Természetesen a fenti listát ízlés szerint bővíthetjük, miután elolvastuk a dokumentációt. Futassuk le.

Létre kell hoznunk egy indító script-et is. (Figyeljünk arra is, hogy a fájl futtatható legyen!) `/etc/init.d/lids:`

```
#!/bin/sh
echo "Starting LIDSadm..."
/sbin/lidsadm -I -- -CAP_SYS_MODULE -CAP_SYS_RAWIO -CAP_SYS_ADMIN \
-CAP_SYS_PTRACE -CAP_NET_ADMIN -CAP_SYS_TIME -CAP_SYS_RESOURCE \
-CAP_LINUX_IMMUTABLE -CAP_KILL -CAP_SYS_TTY_CONFIG \
+INIT_CHILDREN_LOCK +RELOAD_CONF
```

Helyezzünk el indító linket az `/etc/rc[2-5].d` szintekre. Lényeg, hogy ez legyen a legutoljára elinduló démon, ezért adjunk neki *S99*, vagy *Sz* előtagot. Ha minden jól ment, indítsuk újra a gépet és válasszuk a LIDS-t tartalmazó kernelt.

Ha minden jól ment, és nem akadályoztuk meg a rendszer normális indítását, akkor jelentkezünk be *root*-ként és adjuk ki a `lidsadm -v` parancsot. Ez kilistázza a jelenleg érvényes szigorítások beállításait. Ha lokálisan az adott shell-re és gyermekfolyamataira ki akarjuk kapcsolni a védelmet (hogy tudjunk változtatni, karbantartani), adjuk ki ezt a parancsot: `lidsadm -S -- -LIDS` Ekkor a jelszó megadása után „hagyományos” módon tehetjük dolgunkat.

Ha a védett fájloknak megváltozik a helye a lemezen, adjuk ki a `lidsadm -U` parancsot, mely frissíti a konfigurációs állományt.

Mindenesetre teszteljük le a rendszer működését: jelentkezünk be *root*-ként, majd csinálunk tiltott dolgokat és figyeljük, naplózza-e a rendszer tevékenységünket.

Érdemes továbbá rendszeresen frissíteni a LIDS szoftverünket, mert igaz, már közel van az 1.0-s verzióhoz, még sok hibát javítanak ki rajta.

8. A „GNU Free Documentation License”

GNU Free Documentation License

Version 1.1, March 2000

Copyright (C) 2000 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other written document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you".

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (For example, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, whose contents can be viewed and edited directly and straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic

translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup has been designed to thwart or discourage subsequent modification by readers is not Transparent. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML designed for human modification. Opaque formats include PostScript, PDF, proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies of the Document numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a publicly-accessible computer-network location containing a complete Transparent copy of the Document, free of added material, which the general network-using public has access to download anonymously at no charge using public-standard network protocols. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has less than five).

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this

License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section entitled "History", and its title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. In any section entitled "Acknowledgements" or "Dedications", preserve the section's title, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section as "Endorsements" or to conflict in title with any Invariant Section.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections entitled "History" in the various original documents, forming one section entitled "History"; likewise combine any sections entitled "Acknowledgements", and any sections entitled "Dedications". You must delete all sections entitled "Endorsements."

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, does not as a whole count as a Modified Version of the Document, provided no compilation copyright is claimed for the compilation. Such a compilation is called an "aggregate", and this License does not apply to the other self-contained works thus compiled with the Document, on account of their being thus compiled, if they are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one quarter of the entire aggregate, the Document's Cover Texts may be placed on covers that surround only the Document within the aggregate. Otherwise they must appear on covers around the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License provided that you also include the original English version of this License. In case of a disagreement between the translation and the original English version of this License, the original English version will prevail.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright (c) YEAR YOUR NAME.

Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.1
or any later version published by the Free Software Foundation;
with the Invariant Sections being LIST THEIR TITLES, with the
Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.
A copy of the license is included in the section entitled "GNU
Free Documentation License".

If you have no Invariant Sections, write "with no Invariant Sections" instead of saying which ones are invariant. If you have no Front-Cover Texts, write

"no Front-Cover Texts" instead of "Front-Cover Texts being LIST"; likewise for Back-Cover Texts.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

Melléklet

Itt néhány logót mellékelek, hogy lássuk, milyen jelképük van ezeknek a programozóknak.

A GNU logója egy kafferbivaly¹⁹³:

A Linux hivatalos logója ez a pingvin:

A Debian hivatalos logója ez a csiga:

¹⁹³ Mivel angolul a „gnu” kafferbivalyt jelent.

Gyakoriak a Web-oldalak alján
feltüntetett „reklám” logók:

Az Apache hivatalos logója ez a toll:

A Postfix logója a rendszer felépítését
ábrázoló folyamatábra kicsinyített
változata:

A Webalizer program logója:

Az OpenSSH logója egy túskehal:

És a Tripwire logója: