Naprakésznek maradni az épelméjűség határain belül

Hogyan segítenek a SuSE YaST2, a Red Hat up2date és a Debian apt-get eszközök rendet tartani a biztonsági foltok erdejében?

A biztonság mozgó célpont, könnyen elszédül az ember, ha követni próbálja. A Linuxhoz hasonló bonyolult rendszereken az egyik legunalmasabb és legfárasztóbb biztonsági feladat az alkalmazások, a parancsok és a programkönyvtárak naprakész követése. Bárki tanúsíthatja, aki olvassa a Bugtraq-listákat, hogy állandóan kihasználható programhibákat fedeznek fel, és javítófoltokat adnak közre. Egy átlagos Linux-terjesztés többszáz csomagját tekintve senki sem remélheti, hogy lépést tud velük tartani.
Ez nem hangzik túl jól, de sajnos igaz. Mégha módunkban is áll figyelemmel kísérni az összes létező biztonsággal foglalkozó levelezőlistát, és minden egyes közzétett sebezhetőségi pontot befoltozunk, előbb-utóbb mi magunk is egy hibajelentés főszereplőjévé válhatunk. Bizonyos hibák addig nem kerülnek napvilágra, amíg valaki ki nem használja őket.
Vigasztalásul elmondhatjuk, hogy ha felzárkózni nem is lehet teljesen, minden előrehaladás sikerként könyvelhető el. Ha egy háromhetes programhiba miatt törik fel a rendszerünket, még mindig menőbbek vagyunk, annál, akit egy hároméves hibával találnak meg (másrészről akárhogy is nézzük, a betörés az betörés).
Félretéve a tréfát: tisztán statisztikai alapon is belátható, hogyha kevesebb foltozatlan hiba van, kevesebb a sebezhető pont. Annak ellenére, hogy mennyire hálátlan és végeláthatatlan feladatnak tűnik, érdemes megpróbálkozni Linux-telepítésünk naprakészen tartásával. Szerencsére a népszerű Linux-terjesztések újabb változatai tartalmaznak olyan programokat, amelyek a feladat nagy részét elvégzik helyettünk. Ilyenek a SuSE-ben a YaST2, a Red Hatben az `up2date` és a Debianban az `apt-get` (ezek közül egyesek még biztonságosak is!).

Csomag vagy forrás?

A legfontosabb tanáccsal kezdeném, amire sokévnyi kételkedés után jöttem rá: amikor csak lehet, használd a terjesztésed által támogatott csomagot. Ez sok ember számára talán természetesnek hangzik – miért fordítanánk le valami forrásból, ha nem muszáj? A régi motorosok viszont mindent forrásból fordítanak, számukra ez a megszokott, és ehhez értenek. Ez azért alakult így, mert az 1990-es évek elején sokkal kevesebb program volt Linux alatt, mint ma, így sok mindent olyan forrásból kellett fordítani, amit eredetileg más felületekre fejlesztettek. Akkoriban a Linux terjesztésekbe való csomagolásának módszerei sem voltak még annyira fejlettek, kevesebb volt a terjesztés is, és közel sem változtak ilyen gyorsan.
(Kellett-e már valaha forrásból fordítani a `ps` parancsot, mert vadonatúj rendszermagod nem képes együttműködni a terjesztéseddel együtt érkezett régebbi `ps`-változattal? Néhányunk átélte ezt a felemelő élményt. Ezenkívül minden nap 15 kilométert kellett gyalogolnunk az iskoláig kígyóktól hemzsegő mocsarakon át stb.)
Ma már másként mennek a dolgok. Ne érts félre, nem mondom, hogy legyünk a bináris csomagok rabjai. Megeshet, hogy egy alkalmazás olyan tulajdonságára van szükséged, amely hiányzik a terjesztésben szereplő változatból, vagy a saját ízlésed szerint szeretnéd lefordítani a programot, mert nem felel meg a terjesztésben szereplő egyencsomag. Ennek ellenére az esetek többségében sok fontos előnye van a bináris csomagok használatának.
Az első a kényelem: sok alkalmazás egyetlen helyről való letöltése gyorsabb és könnyebb, mint ha egyesével kellene leszedegetni őket a fejlesztők honlapjairól (főként ezért alakultak ki a terjesztések), és a bináris csomagok telepítése sokkal gyorsabb és hibátlanabb, mint a fordítás. A kényelem nem az az érték, amit mi, vakbuzgó unixosok különösebben elismerünk, de azért el sem hanyagolandó.
A második előny a csomagok megbízhatósága, ugyanis a nagyobb terjesztéseknél a csomagolás előtt az alkalmazásról eldöntik, hogy érdemes-e betenni a terjesztésbe, és ha igen, melyik a legmegbízhatóbb változata, milyen fordítási beállítások illenek legjobban a terjesztéshez stb. Volt már ez alól egy pár híressé vált kivétel, manapság azonban a legtöbb terjesztésnél elég rendesen ellenőrzik a termék minőségét.
A megbízhatóság magától értetődően a biztonság szempontjából is fontos: ahol hibák vannak, ott sebezhető pontok is akadnak. Még az olyan hibákat is ki lehet használni, amelyeknek nincs nyilvánvaló biztonsági következményük, gondoljunk csak a szolgáltatásmegtagadásos (DoS-típusú) támadásokra, amelyeknek az a célja, hogy lefagyasszák a rendszert.
Ez elvezet az alkalmazások biztonságának egyik kiábrándító ellentmondásához: bár az Interneten leginkább elterjedt sebezhetőségek a gyengén karbantartott alkalmazásokból erednek (elavult, illetve ismert hibával bíró változatok), az újabbak nem feltétlenül jobbak. A biztonsággal foglalkozó szakemberek az évek során joggal marasztalták el a Microsoftot, hogy csigalassúsággal ismeri el a hibáit, és termékeihez ráérősen adja ki a javításokat. De ugyanúgy keserű tiltakozás követi azt is, amikor a Microsoft gyorsan ad ki egy a megbízhatóságot rontó javítófoltot, mert ez nagymértékben csökkenti az eredeti hiba kijavításával elért előnyt.
Most egy pillanatra hagyjuk figyelmen kívül, hogy a megbízhatóság önmagában is kívánatos, és gondolkodjunk el a következőkön: tegyük fel, hogy egy alkalmazásban lehetőség kínálkozik egy jelentéktelen tártúlcsordulási hiba elméleti kihasználására a rendszergazdai jogok szerzésének céljából, de ezt csak olyan felkészült assembly-programozók tudják kiaknázni, akik az RC4-folyamok titkosításához is jól értenek. Ezt egy olyan kódrészlettel foltozzuk meg, ami miatt lehetőség nyílik a gép lefagyasztására, amit még a jobb elfoglaltságot nem találó iskolás kölykök is ki tudnak használni. Nagyobb lett ettől a biztonság? A válasz a pontos körülményektől függ és attól, hogy kinek tesszük fel a kérdést, de a tanulság az, hogy a programfrissítések gyakran ilyen gondokat vetnek fel.
Talán túlmagyarázom ezt a témakört, de fontos, hogy eloszlassuk azt a tévhitet, miszerint az azonnali programfrissítés valamiféle csodaszer. Évekig tartott, míg teljesen megértettem, hogy például miért csomagolják a legeslegújabb OpenBSD-

terjesztésbe is a BIND v.4.9.8-at, ami számítógépévekben mérve már ősöregnek számít.
Azért is fontos ez – eredeti témánkhoz visszatérve –, hogy megértsük, miért célszerű ráhagynunk Linux-terjesztésünk összeállítóira a nehéz foltozási döntések meghozatalát, és a foltozással a terjesztés által kiadott hivatalos (és remélhetőleg kipróbált) javítást megvárnunk. Épp elég nehéz lépést tartani a terjesztés által kiadott biztonsági frissítésekkel – teljesen kilátástalannak tartom, hogy saját magam foltozzam és fordítsam újra az összes lényeges alkalmazást, és izgatottan várjam, hogy a foltozással nem rontottam-e el valami mást is.
Összegezve az eddigi bölcsességeket: naprakésznek lenni erény, bináris csomagokat használni a lustaság erényes formája, és a másodkézből származó (a Linux-terjesztésed által kiadott) biztonsági frissítések használata az egyéni „vagánykodás" helyett nem lustaság, hanem okosság.

Melyik rendszert kell frissíteni és hogyan?

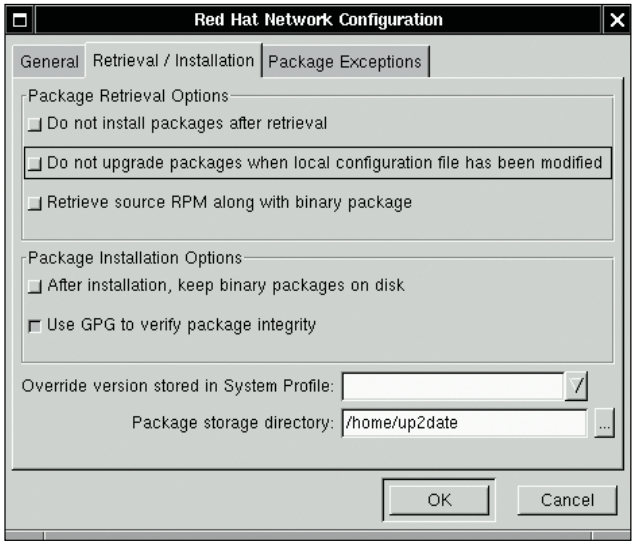
Mielőtt az önműködő frissítési eszközökről szólnék, beszéljünk egy kicsit a rendszerek különféle fajtáiról. Az Internetre közvetlenül csatlakoztatott kiszolgálónak sokkal sürgetőbb igénye van az azonnali alkalmazásfrissítésre, mint egy tűzfal mögött elhelyezett asztali gépnek. A józan ész segít annak a döntésnek a meghozatalában, hogy melyik rendszernél kell több időt fordítani a szigorú naprakészen tartásra, és melynél fogadható el alacsony kockázati szint a kisebb energiaráfordítás következtében.
A másik dolog, amit a rendszerek szerepe kapcsán meg kell fontolni: melyik gépen futtatható kényelmes X-alapú frissítőprogram, mint amilyen például a YaST2? Az Internetre kapcsolt kiszolgálón nem tanácsos X-kiszolgálót futtatni. Ha nagyon tetszik a YaST2 könnyű használhatósága és az általa biztosított kényelem, futtasd egy belső rendszeren, állítsd be a frissítések mentését, és a frissítéseket `scp`-vel másold fel a külső gépre.

Red Hat up2date

A Red Hat 6.2 óta az `up2date` program használható a frissített csomagok önműködő azonosítására, letöltésére és telepítésére. Az `up2date` szöveges és grafikus módban is futtatható, így a „Hol futtathatom?" fent taglalt kérdése fel sem merül nála.
Az `up2date`-et a futtatás előtt be kell állítani. Szerencsére két egyszerű és kényelmes eszköz is a segítségünkre van. Először létre kell hozni egy felhasználói fiókot a Red Hat Networkön (RHN) az `rhn_register` használatával. Kapcsolók nélkül indítva az `rhn_register` grafikus módban indul, de a `nox` kapcsolót megadva szöveges módban is futtatható::

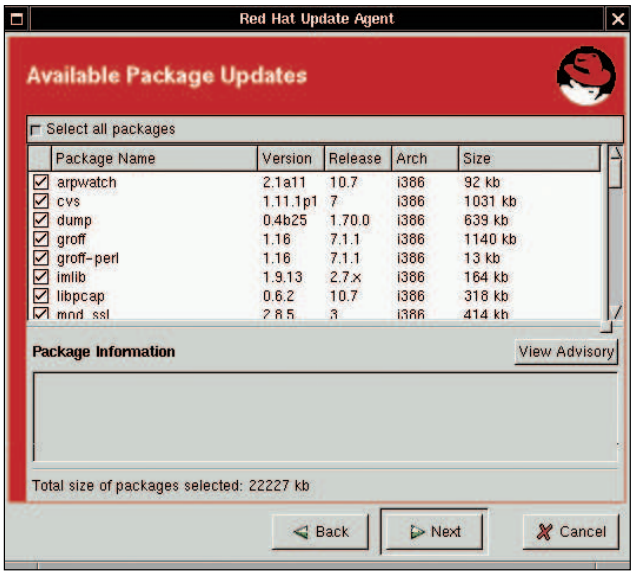
```
bash-# rhn_register -nox
```

Az RHN-nél egy rendszer díjmentesen bejegyeztethető, azonban az ugyanahhoz a fiókhoz tartozó további rendszerek után díjat kell fizetni. RHN-bejegyzés szükséges minden olyan gépre, amelyen az `up2date`-et futtatni akarod, viszont semmi sem akadályoz meg benne, hogy egyetlen gépen futtasd az `up2date`-et, majd mentsd a frissített csomagokat, és többi Red Hat rendszeredre `scp`-vel vagy más biztonságos módon átmásold, majd a rendszereken kézzel telepítsd őket.
Egy további dolog, amit tudnod kell az RHN-ről, hogy az RHN-be bejegyzett rendszer alapértelmezés szerint a számítógép hálózati beállításait, alkatrészeinek listáját és az összes telepített csomag nevét, valamint változatszámát elküldi az RHN-nek. Ennek segítségével ütemezheted az RHN által küldött önműködő frissítéseket, illetve testreszabott leveleket kaphatsz, ha a gépedre telepített valamelyik csomaghoz frissítés jelent meg.
Valószínű, hogy ez a két lehetőség tetszeni fog neked, és amennyiben a kettő közül bármelyik is megkönnyíti a rendszeres biztonsági frissítések kezelésével kapcsolatos munkádat, nem foglak bírálni érte. Nekem azonban nincs ínyemre, hogy idegeneknek részletes listát adjak a gépem beállításairól. Nem arról van szó, hogy okom lenne a bizalmatlanságra a Red Hatnél dolgozó remek szakemberekkel szemben, csak azt állítom, hogy nem kell megbíznom bennük, ha nincs kedvem hozzá.
Végül is nem olyan nagy feladat a Redhat-Watch listát járatni, amelyen a Red Hat bejelenti a frissítéseket (➲ http://listman.redhat.com), és ezután eldönteni, hogy futtatni kívánom-e az `up2date`-et. Ha szükséges, az `up2date` akkor is megállapítja, hogy mely frissítések érintik a rendszeremet, ha a Red Hat semmilyen adatot nem tárol rólam. Emiatt én azt a gyakorlatot követem (és mindenkinek ezt javaslom), hogy ne jelöljük be az `rhn_register` következő beállításait: ***Include information about hardware and network*** (Az alkatrészek és a hálózat adatainak továbbítása) és ***Include RPM packages installed on this system in my System Profile*** (Telepített RPM-csomagok feljegyzése a felhasználó rendszerprofiljában).

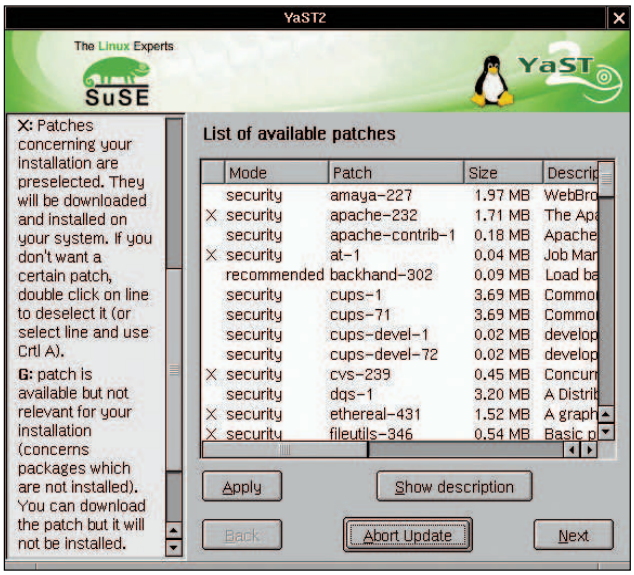


1. kép A Red Hat up2date-config eszköze

Miután rendszeredet bejegyeztetted a Red Hat Networknél, futtatnod kell az `up2date-config` programot (lásd az *1. képet*). Az `rhn_register`-hez hasonlóan ez a parancs is támogatja a `nox` kapcsolót, ha szöveges módban kívánod futtatni. Az `up2date-config` használata többnyire magától értetődik, de egynéhány beállítást érdemes megemlíteni.
Először is: ha a fent leírt módon egy központi gépet szeretnél használni a frissítések tárolására és terjesztésére, az ***After installation, keep binary packages on disk*** (Telepítés után a binárist hagyja a lemezen) beállítást be kell jelölnöd, és a ***Package storage directory:*** (A csomagok tárolási helye) alapértelmezett értékét is meg kell adnod (vagy legalább megjegyezned). Ha az `rhn_register`-t a `-nox` kapcsolóval futtattad, ezek a beállítások ***keepAfterInstall*** és ***storageDir*** néven szerepelnek.



2. kép Az up2date egyik képernyője



3. kép Frissítés a YaST2-vel

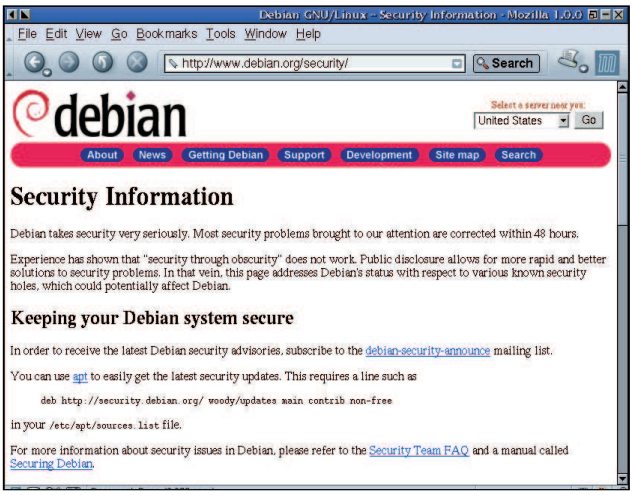
Másodszor: hacsak nem túl lassú a géped, ne tiltsd le az alapértelmezés szerint engedélyezett ***Use GPG to verify package integrity*** (GPG használata a csomag épségének ellenőrzésére) beállítást. Az rpm formátum egyik legjobb tulajdonsága a belső GPG-aláírás használata, amellyel ellenőrizhető a csomag épsége. Az `up2date` ezt az ellenőrzést magától elvégzi, ha a GnuPG telepítve van a gépen. Egyénként az RPM-csomagok GPG-aláírását a következő paranccsal kézzel is ellenőrizheted:

```
rpm --checksig /path/packagefilename.rpm
```

Természetesen a */path/packagefilename.rpm* helyére az ellenőrzendő RPM-fájlt kell írni, a teljes elérési úttal.
Előbb vagy utóbb valaki feltöri a Red Hat egyik tükörkiszolgálóját, és valamelyik létfontosságú csomagot egy trójai falóval helyettesíti. Ha ez megtörténik, akkor azokat, akik rendszeresen ellenőrzik az RPM-csomagok aláírását, ez a gond sokkal kevésbé fogja érinteni.

Milyen gyakran frissítsd az egész terjesztést?

Tegyük fel, hogy a képzeletbeli Bo-Weevil Linux 33.1 terjesztést használod, és mindig elvégezted a szükséges csomagfrissítéseket. Mi a teendő, ha megjelenik a Bo-Weevil 33.2? A folyamatos naprakész állapot fenntartásához szükséges-e a 33.2-re történő frissítés?
Lehetséges, de valószínűleg nem. A terjesztés frissítése (a csomagfrissítésekkel szemben) általában új tulajdonságokat, csomagokat és igen, biztonsági frissítéseket is ad a rendszerhez, azonban a terjesztés frissítése önmagában nem biztosítja, hogy a rendszer biztonságosabbá válik. Más szavakkal kifejezve: a teljesen megfoltozott Bo-Weevil 33.1 valószínűleg biztonságosabb, mint a foltozatlan 33.2, és legalább olyan biztonságos, mint a foltozott 33.2-es rendszer.
Csak akkor frissítsd a terjesztést, ha olyan új tulajdonságokkal rendelkezik, amelyekre szükséged van. Azonnal frissíts, ha az általad használt változat támogatása hivatalosan megszűnik (például több biztonsági frissítést nem adnak ki). Ha egyik eset sem áll fenn, továbbra is frissítgesd a régebbi terjesztés csomagjait, és ne ess a változatszámhóbort hibájába.

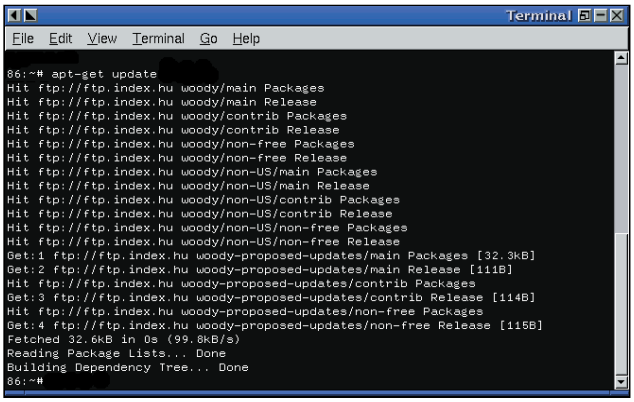


4. kép A Debian/GNU biztonsággal foglalkozó weboldala

Miután bejegyeztetted magad az RHN-nél, és lefuttattad az `up2date-config`-ot, futtathatod magát az `up2date` programot (lásd a *2. képet*). Erről nem lehet túl sokat mondani, az `up2date` célja eleve az egyszerűség és a kényelem, szükségtelen ecsetelnem, hogyan kell a világosan feliratozott gombokra kattintgatni.
Megismétlem korábbi tanácsomat: iratkozz fel a Redhat-Watch listára, és amikor a rendszeredet érintő frissítést bejelentik, azonnal futtasd az `up2date`-et. Egy ilyen remek és felhasználóbarát eszköz birtokában nincs mentséged, ha elmulasztod.

A SuSE frissítőrendszere

Ha SuSE Linuxot használsz, még egyszerűbb a dolgod. Az önműködő frissítésekhez a YaST2 frissítőmodulját használhatod (lásd a *3. képet*).



5. kép Az apt-get update parancs kimenete

A Red Hat `up2date`-tel ellentétben a SuSE-nél a számítógépet nem kell bejegyeztetni a program futtatása előtt, és külön alkalmazás vagy fájl segítségével beállítani sem szükséges.
A frissítőmodul mindent egyben tartalmaz. Az első néhány képernyőjén szükség esetén megváltoztathatóak a beállítások (például a csomagok letöltéséhez a tartózkodási helyedhez közelebbi kiszolgálót is választhatsz, mint például a SuSE amerikai ➲ ftp.suse.com kiszolgálója).
Ha tudni akarod, mikor kell elindítani a frissítőmodult, iratkozz fel a SuSE suse-security-announce levelezőlistára. A Redhat-Watch listához hasonlóan a forgalom kicsi, és ne aggódj amiatt, hogy a SuSE komolytalan üzenetekkel szemeteli tele a postaládádat. A feliratkozáshoz látogass el a ➲ http://www.suse.com/en/support/mailinglists/index.html címre.
Őszintén szólva sokkal többet nem kell elmondanom a YaST2 frissítőmoduljáról, kivéve egy apró hibát, amit tapasztaltam. Tulajdonképpen a felhasználó hibája, de könnyű elkövetni.
Ha a `yast2` parancsot egy `xterm`-ből, vagy a ***Parancs futtatása*** párbeszédablakon keresztül kézzel adod ki, és az X nem rendszergazdaként fut, a frissítés sikertelen lesz, és azt a félrevezető hibaüzenetet kapod, hogy a frissítési lista a megadott FTP-kiszolgálón nem érhető el.
Nem ez az igazi ok, valójában a YaST2 rendszergazdai jogosultságokat igényel ahhoz, hogy ezt a fájlt kiírja a lemezre, miután az FTP-kiszolgálóról letöltötte. Ez nem azt jelenti, hogy a YaST2csak a rendszergazda által futtatott X-munkafolyamatból indítható, de ha az X nem így fut, akkor a YaST2-t a SuSE által létrehozott menübejegyzés segítségével kell elindítani, mert ebben az esetben elkéri a rendszergazda jelszavát, és a frissítés rendben megtörténik.

Az RPM-csomagok kézzel történő frissítése

Annyira fellelkesített az `up2date` és a YaST2, hogy még nem is említettem az RPM-fájlok frissítésének egyik legegyszerűbb módját: magát az `rpm` parancsot. Ez a módszer ugyanolyan jól működik a Red Hat (és származékai), valamint a SuSE alatt. Használatát egy példán keresztül mutatom be.
Tegyük fel, hogy értesítést kaptál a képzeletbeli SuSE vagy Red Hat „blorpflap” csomag sebezhetőségéről és az azt megszüntető frissítésről. A frissített RPM-csomagot az értesítésben megadott címről letöltötted a helyi ***/usr/pkg/updates/blorpflap-3.2-3.rpm*** útvonalra. Először ellenőrizd a csomag érvényességét:

```
rpm --checksig /usr/pkg/updates/
➥blorpflap-3.2-3.rpm
```

Természetesen ehhez az szükséges, hogy a terjesztés csomagaláíráshoz használt kulcsa rajta legyen a nyilvános GnuPG kulcscsomódon (a Linuxvilág 2001. októberi és novemberi számában megjelent egy kétrészes írásom a GnuPG használatáról).
Miután a GPG-aláírás ellenőrzése rendben zajlott (vagy feltételezed, hogy rendben zajlana le, amit a saját felelősségedre megtehetsz – a fenti lépés nem kötelező), telepítsd a frissítést:

```
rpm -Uvh /usr/pkg/updates/blorpflap-3.2-3.rpm
```

A `-U` természetesen az update (frissítés) rövidítése (valójában frissítés vagy telepítés), és már telepített csomagok frissítésére vagy új csomagok telepítésére használható; a `-v` bőbeszédűvé teszi a folyamatot; a `-h` hatására pedig a folyamat előrehaladásáról képet kaphatunk.

A Debian apt-get programja

Ebben a hónapban ez az utolsó eszköz, amit bemutatok.
A Debian általános jellegzetességének megfelelően az `apt-get` kevésbé csillog-villog, csakhogy bizonyos szempontból könnyebben használható, mint a többi terjesztés díszes grafikus frissítőeszközei. Dióhéjban elmondva: a `deb`-csomagok frissítése a Debian rendszeren két lépésből áll:

1. a csomaglista frissítése,
2. az új csomagok letöltése és telepítése.

Mindkét lépés az `apt-get` segítségével hajtható végre:

```
bash-# apt-get update
bash-# apt-get -u upgrade
```

A második parancs hatására az `apt-get` a `wget` segítségével letölti a frissített csomagokat és telepíti őket.
A Debian biztonsági hibáiról és a frissítésekről értesítést kaphatsz, ha feliratkozol a debian-security-announce levelezőlistára a ➲ http://www.debian.org/MailingLists/subscribe oldalon található űrlap kitöltésével. Futtasd le az `apt-get` programot, ha értesítést kapsz egy a rendszeredet érintő biztonsági frissítésről.
Bármennyire is szeretem az `apt-get` programot, akad egy fontos hiányossága: nem lehet vele GPG-aláírásokat ellenőrizni. Ez azért van így, mert a deb formátum sajnos nem támogatja az aláírásokat, és a Debian-csomagokat jelenleg külső aláírással sem látják el. Állítólag a deb formátum következő változata már támogatni fogja a GPG-aláírásokat.

Ennyi jutott erre a hónapra. Sok szerencsét kívánok a frissítésekhez!

Linux Journal 2002. július, 99. szám

Mick Bauer
(mick@visi.com) hálózati biztonsági tanácsadó az Upstream Solutions Inc.-nél Minneapolisban (Minnesota). Mick a szerzője a hamarosan megjelenő új O'Reilly-könyvnek, amelynek címe „Building Secure Servers With Linux”, de ő írta a „Network Engineering Polka” című művet is. Büszke apja gyermekeinek.