GPG, a legjobb szabad titkosító program (1. rész)

Bevezető az alulértékelt, százszázalékosan szabad segédeszköz használatába, amelyről nem tudtad, hogy szükséges (pedig az) – a GnuPG.

Tíz év telt el azóta, hogy ***Phil Zimmermann*** megjelentette a PGP v1.0-t (Pretty Good Privacy). Az ekkor még üldözési mániások titkos eszközének tartott PGP mára a levelezés titkosításának legelfogadottabb szabványává – mi több, internetes szabvánnyá – vált. A GnuPG (GNU Privacy Guard) a kereskedelmi forgalomban kapható PGP szabadon felhasználható változata, és a legtöbb Linux-terjesztés tartalmazza. Ennek ellenére mégsem használják annyian, amennyien megtehetnék.
Te is azok közé tartozol, akik idegenkednek a GnuPG használatától? Ez a kétrészes cikk remélhetőleg téged is meggyőz alkalmazásának fontosságáról. Miután legyártottad a személyes kulcsaidat és elküldted az első titkosított leveledet, továbbá ellenőrizted annak a szuper jó programnak a biztonsági aláírását, amelyet az imént töltöttél le, örülni fogsz, hogy elfogadtad a kihívást és megismerkedtél ezzel a többszolgáltatású csodával, a GnuPG-vel.
Az első részben a PGP/GnuPG hátterével, a hozzá kapcsolódó fogalmakkal foglakozunk, majd rátérünk a gyakorlati alapokra. A második részben részletesebben tárgyaljuk a fájl- és levéltitkosítást, a kulcskezelést és a grafikus felhasználói felületeket.

Néhány szó a PGP történetéről, és arról, hogy miért nem a PGP-ről írok

1991-ben, amikor az amerikai kongresszus arra készült, hogy törvénytelennek minősítse a titkosító programok magáncélú használatát, Phil Zimmermann megjelentette a PGP v1.0-t. Ez az eredetileg szabadon felhasználható eszköz tette lehetővé a számítógép-felhasználók számára, hogy titkosítsák a személyes adataikat és a levelezésüket. Ez annyira hatékonynak bizonyult, hogy a legelszántabb és biztos anyagi háttérrel rendelkező kíváncsiskodók számára is fejtörést okozott (ide sorolandó például az Egyesült Államok kormánya).
Phil Zimmermann története fontos és magával ragadó, elolvashatod ***Simon Garfinkel*** könyvében, amely Phil weblapján (lásd ***Kapcsolódó címek***) is megtalálható. Most legyen elegendő annyi, hogy a kormány nyomozása, a szabadalmi bonyodalmak és a céges felvásárlások ellenére a PGP tovább fejlődött –, Zimmermann elképzeléseinek megfelelően szolgálva a magánszemélyek és céges felhasználók igényeit világszerte.
Amikor ezt leírom, a tágabb értelemben vett PGP-re gondolok, amely a PGP-n kívül az OpenPGP-t, valamint a GnuPG-t is jelenti. Ez utóbbi megjelenése eredményezte a PGP kulcs- és üzenetformátumának internetes szabványként való elfogadását (az RFC 2440 rögzíti). Ezáltal a PGP teljesen szabad megvalósítása minden felhasználó számára elérhetővé vált a világon.
Annak ellenére, hogy Zimmermann kétségkívül a nyílt forráskód egyik igazi úttörője, a Network Associates Inc. (NAI) által terjesztett PGP általában gondot okozhat a nyílt forrás híveinek, a linuxosok számára pedig különösképpen. A legszembetűnőbb nehézség, hogy a kereskedelmi forgalomban kapható PGP csak Windowson és Mac OS-en fut. Másodsorban a PGP Freeware is csupán a nem üzleti felhasználók számára ingyenes, azaz az oktatásban dolgozók és a nem haszonelvű szervezetek számára. Harmadsorban a NAI nagymértékben csökkentette a PGP forrásának azon részét, amelyet kódátvizsgálás céljából nyíltan elérhetővé tesz.
Ez utóbbi fejlemény eredményeképpen Phil Zimmermann lemondott a PGP Securitynél betöltött tisztségéről. Felvetődik a kérdést, hogy vajon megbízhatunk-e a PGP NAI által terjesztett változatában? Ismervén az amerikai kormány titkosítással kapcsolatos kedvezőtlen álláspontját és a próbálkozásaikat, hogy a titkosítással foglalkozó programokba hátsó kapukat építtessenek, túlságosan könnyű elképzelni, hogy a NAI enged a nyomásnak, és valóban létrehoz ilyen hátsó bejáratokat.
Mivel a nyilvánosság számára a PGP forrása nem érhető el teljes egészében, nem áll módunkban ellenőrizni a NAI állításait, melyek szerint a program biztonságos.
A GnuPG ezzel szemben teljesen nyílt forrású és ingyenes programcsomag, nagyjából ugyanazt nyújtja, mint a PGP (hiányzik belőle a VPN-támogatás és a lemezkötet-titkosítás – ezek azonban megtalálhatók a PGP Desktopban). Rövid idő alatt a GnuPG lett a Linux-felhasználók legkedveltebb levél- és adattitkosító programja, hiszen minden Linux-terjesztésben fellelhető. A GnuPG fejlesztés alatt áll, melynek vezető fejlesztője ***Werner Koch***.

Mi az a GnuPG és miért van rá szükséged?

A GNU Privacy Guard egyetlen futtatható állományból áll, a *gpg*-ből. Valójában a csomag még egy ***gpgv*** nevű programot is tartalmaz, de ez a program a gpg szolgáltatásainak kényelmi szempontok szerint kiválogatott elemeit öleli fel, így bátran állíthatjuk: a futtatható gpg mindent tartalmaz. Így lehetővé válik, hogy felváltva alkalmazzam a gpg és GnuPG kifejezéseket – ezt ki is használom a cikk hátralevő részében. A PGP kifejezést is tágabb értelmében fogom használni: nemcsak a NAI által forgalmazott termékre, hanem a PGP, az OpenPGP és a GnuPG közös protokolljára, eljárásaira és a bizalomhálóra (Web of Trust) is.
A GnuPG több kiegészítő szerep mellett négy főbb feladatot lát el: az adattitkosítást, a titkosított adatok megfejtését, a digitális aláírás kezelését és ellenőrzését. A program segítségével létrehozhatjuk és kezelhetjük a titkosításhoz szükséges kulcsokat – ezek olyan tevékenységek, amelyek nem játszanak közvetlen szerepet a titkosításban, mégis elengedhetetlenek.
Egyszerűbben fogalmazva a GnuPG-t fájlok – kifejezetten

Igény az igazi bizalomra

Széljegyzet

Képzeljük el a következőt: valaki létrehoz egy hamis kulcspárt egy programfejlesztő nevében, feladja a nyilvános kulcsot pgp.mit.edu-ra (egy népszerű kulcskiszolgáló), majd a programcsomagot egy vírusokat tartalmazó csomagra cseréli ki, és létrehozza a megfelelő aláírást. Mi akadályozza meg ebben? Semmi. A Tripwire-höz hasonlóan a PGP nem gátolja meg a hamis fájlok létrehozását, csak figyelmeztet.

Ha a csomagot és a hozzá tartozó aláírást is meghamisították, jó esetben a következők egyike történik:

- Amikor ellenőrizni próbálod az aláírást, a GnuPG figyelmeztetni fog, hogy az a kulcs, amivel az aláírást létrehozták, eltér attól, amit a legutóbb használtak (erre a figyelmezetésre akkor nyílik lehetőség, ha már rendelkezel az eredeti nyilvános kulccsal).
- Ha még nincs a nyilvános kulcsokat tartalmazó kulcscsomón a fejlesztő nyilvános kulcsa, észre fogod venni, hogy a kulcs azonosítója nem egyezik meg azzal az azonosítóval, amit az igazi fejlesztő használ a fórumokra írt leveleiben vagy a webkiszolgálóján. (Igen, lehet, hogy ehhez egy kicsit keresgélni kell a Weben).
- Valaki más már észrevette a kulcsok körüli furcsaságokat, és a gondot elhárítják, még mielőtt megpróbálnád ellenőrizni az aláírást.

Nagyon könnyen előfordulhat, hogy a fent felsoroltak közül sajnos egyik sem történik meg. A fentiekben ecsetelt csalás után nagy valószínűséggel a következő események várhatók: néhány felhasználó letölti a trójai falovat, és a legtöbbjük meg sem próbálja ellenőrizni az aláírást. Ők nekifognak és azonnal telepítik a hamis programot. Rajtuk kívül még sok felhasználó letölti a programot, majd a hamis aláírás által megkövetelt hamis nyilvános kulcsot, ellenőrzik az aláírást, majd boldogan telepítik a hamis programot.

Szerencsére az is elegendő, ha csupán egyetlen felhasználó (aki rendelkezik a fejlesztő eredeti kulcsával) észreveszi a csalást és megkongatja a vészharangot. Remélhetőleg elég korán teszi meg ahhoz, hogy a kevésbé figyelmes felhasználók hamisítványtelepítése megakadályozható legyen. A történetből két tanulságot is levonhatunk: kiemelkedően fontos, hogy a lehető legtöbb felhasználó alkalmazza is a GnuPG-t az aláírások ellenőrzésére, valamint azt is meg tudják állapítani, hogy az aláírást érvényes kulccsal hozták-e létre.

Véleményem szerint ez megkérdőjelezi azt a gyakorlatot, hogy a programcsomagot, az aláírását és az aláíró nyilvános kulcsát ugyanazon a kiszolgálón tartják. Az egész kiszolgálót sem nehezebb meghamisítani, mint a részeit.

Utolsóként álljon itt ismét egy példa a központi szervezetekbe vetett bizalom buktatóiról. Emlékezzünk, hogy márciusban a Microsoft bejelentette: két microsoftos digitális aláírás hamisítványnak bizonyult, amelyet a VeriSign által hitelesnek minősített. (A programcsomagok és frissítések hitelesítéshez a Microsoft a saját megvalósítását – a GnuPG/PGP-vel azonos elven működő digitális aláírásokat – alkalmazza).

Ha valaki a hamis tanúsítványokkal aláírt volna egy módosított Internet Explorert, és elérte volna, hogy ez a hamisított csomag a Microsoft weblapjáról letölthető legyen, az álprogramot telepítő felhasználónál megjelent volna a felbukkanó ablak, amely arról tájékoztat, hogy a telepítésre kerülő program a VeriSign szerint biztonságos forrásból származik.

Ez történik, amikor az emberek, különösen az olyan szervezetek, mint a VeriSign, nem fordítanak kellő figyelmet arra, hogy milyen kulcsokban bíznak meg vagy akár írnak alá. A Microsoft a hamis kulcsok létezését csak három hónappal a felfedezésük után jelentette be. Ellenőrizd az aláírásokat – ne bízz vakon minden kulcsban!

levelek – titkosítására és a titkosított üzenetek, valamint mellékletek megfejtésére, továbbá dokumentumok, programforrások és más elektronikus adatok digitális aláírására és az ilyen aláírások ellenőrzésére használják. A digitális aláírás ellenőrzésével megállapíthatjuk, hogy az adott fájlt ténylegesen az írta alá, aki állítja magáról, valamint azt is, hogy a fájl tartalma nem változott-e meg az átvitel során (nem módosították-e rosszhiszeműen). A program segítségével a két legfontosabb kulcscsomót is kezelhetjük: a személyes kulcsokat tartalmazó „titkos kulcscsomót" (secret keyring), és az ismerőseink, munkatársaink, üzletfeleink nyilvános kulcsait tartalmazó „nyilvános kulcscsomót" (public keyring).

A GnuPG nyilvánvalóan akkor szükséges, ha más GnuPG-felhasználókkal (vagy OpenPGP-megfelelő programokat használókkal) szeretnél titkosított üzeneteket váltani. A program lehetőséget nyújt a helyi merevlemezen található fájlok titkosítására is. Ennek akkor lehet értelme, ha például hordozható számítógép merevlemezéről van szó és fennáll az eltulajdonítás veszélye. Ha egyetlen ismerősöd sem használ GnuPG-t vagy PGP-t, és úgy érzed, hogy nincsenek titkosításra szoruló adataid, még mindig létezik egy ok, ami miatt érdemes megismerkedni a GnuPG-vel: ez a programcsomag-aláírás. Néhány forgalmas nyilvános FTP-kiszolgáló feltörésének következménye – amikor a letölthető programcsomagokat hamis „trójai falovakkal" helyettesítették –, hogy a biztonságra valamit is adó programfejlesztők elkezdtek digitálisan aláírt programcsomagokat terjeszteni.

Így működik a nyilvános kulcsú titkosítás

A nyilvános kulcsú titkosítás (PK crypto) alapjairól már esett szó „Az OpenSSH száz meg egy előnye" című írásunkban (Linuxvilág, 2001. február–március, 62. oldal). A témakör azonban annyira lényegbevágó, hogy érdemes vele újra foglalkozni, annál is inkább, mert a GnuPG-vel is kapcsolatos.

Emlékezzünk, a PK crypto alapjai egyszerűek: mindenkinek két kulcsa van, egy nyilvános kulcs és egy személyes titkos kulcs. Egy adott nyilvános kulccsal titkosíthatjuk a nyilvános kulcs tulajdonosának szánt adatokat, valamint ellenőrizhetjük a kulcs tulajdonosa által aláírt fájlokat. Titkos kulcsunkkal fejthetjük meg a hozzánk érkező titkosított üzeneteket, és hozhatunk létre digitális aláírást.

A nyilvános kulcsokat terjesztésre szánják. Akinek a nyilvános kulcsodat odaadod, kódolt üzenetet küldhet neked, és ellenőrizheti digitális aláírásod hitelességét.

Ennek megfelelően a személyes kulcsnak szigorúan titkosnak kell lennie, a jogtalan másolástól és felhasználástól gondosan meg kell védeni. Fontos, hogy a neked szánt adatokat lehetőleg csak te fejthesd vissza, és csak te állíthass ki magadnak hiteles aláírást.

A nyilvános kulcsokat sokféleképpen lehet terjeszteni, ezeknél ugyanis a sértetlenség a fontos, nem a titkosság. Titkos kulcsaidat biztos helyen kell tartanod, például egy olyan könyvtárban, amire csak te rendelkezel olvasási joggal, továbbá hosszú és nehezen megfejthető jelmondatot kell használnod. Ezt a jelmondatot kell megadnod, amikor a titkos kulcsodat használod (nemsokára elárulom, hogyan teheted meg).

```
       PØlda az alÆ rÆs-ellenırzØsre

mick@kolach:~ > gpgv gpa-0.4.1.tar.gz.sig

gpgv: Signature made Thu 05 Apr 2001
09:21:26 AM CDT
      using DSA key ID 621CC013
gpgv: Can't check signature: public key
      not found

mick@kolach:~ > gpg --keyserver pgp.mit.edu
--recv-keys 621CC013

gpg: requesting key 621CC013 from
pgp.mit.edu ...
gpg: key 621CC013: public key imported
gpg: Total number processed: 1
gpg:               imported: 1

mick@kolach:~ > gpgv --keyring pubring.gpg
gpa-0.4.1.tar.gz.sig

gpgv: Signature made Thu 05 Apr 2001
09:21:26 AM CDT
      using DSA key ID 621CC013
gpgv: Good signature from "Werner Koch
<wk@gnupg.org>"
gpgv: Notation: remark=I have not checked
the source.
      It is just the current CVS version
```

A fent leírtak minden PK crypto rendszer esetében megegyeznek. A rendszerek – az SSH, a PGP, illetve a GnuPG és egyéb PK-alkalmazások – közötti különbség a nyilvános kulcsok terjesztésében és hitelességük megállapításában rejlik. Ez nagyon fontos, mert bármilyen gondosan őrzöd a titkos kulcsodat, komoly gondban leszel, ha nem lehet különbséget tenni az igazi és az ellenségeid által terjesztett hamis nyilvános kulcsok között (lásd az ***Igény az igazi bizalomra*** című széljegyzetet).
Az SSH esetében ez a kérdés nem igazán megoldott. Ha Béla barátod szeretne belépni a kiszolgálódra, és ehhez a DSA kulcspárját óhajtja használni, akkor a nyilvános kulcsot elküldi neked levélben. Ezután rajtatok áll, hogyan döntitek el, hogy a kulcs valóban hiteles-e (a legtöbb esetben felhívod Bélát telefonon és beolvasod neki a kulcs egy részét).
A GnuPG és a PGP esetében a kulcsok valódiságának ellenőrzésére léteznek eljárások, de rajtad áll, hogy ezeket a lehetőségeket kihasználod-e. Ezeknek az eljárásoknak az összességét bizalomhálónak nevezzük.

A bizalomháló

A bizalomhálót nem nehéz megérteni. Ha Béla ismeri Tamást és aláírja Tamás kulcsát; bárki, aki ismeri Bélát, megbízhat Tamásban is, tehát ha Béla aláírja Tamás nyilvános kulcsát a saját (Béla) titkos kulcsával, akkor Tamás terjesztheti a Béla által aláírt nyilvános kulcsot. Ha valaki birtokolja Béla nyilvános kulcsát és megkapja Tamás Béla által aláírt nyilvános kulcsát, ellenőrizheti Béla aláírásának valódiságát, ami igazolja Tamás kulcsának valódiságát.
Minél többen írják alá Tamás nyilvános kulcsát, annál nagyobb a valószínűsége, hogy ha valaki megkapja Tamás nyilvános kulcsát, annak valódi kulcsát egy ismert kulcs alapján ellenőrizni tudja. Ugyanez érvényesül azokra is, akik Tamás kulcsát aláírták. Ha Tamás elküldi nekem a nyilvános kulcsát, de nem ismerem se Tamást, se Bélát, akkor Béla aláírása nem hat meg egy csöppet sem. De ha Béla kulcsát aláírta Aladár, aki az én jó barátom, megbízom a kulcsában és rendelkezem vele, megbízhatok Béla kulcsában (hiszen Aladár aláírta) és így Tamás kulcsában is. Innen származik a „háló" elnevezés.
Ez az eljárás lényegesen eltér a ***VeriSign*** és az ***Entrust*** mintájától, ahol egy központi, mindenki által elismert szervezet (PKI – public key infrastructure) hitelesíti az összes nyilvános kulcsot. A bizalomháló – még ha messze is áll a tökéletességtől – hibatűrő rendszer: a bizalom megoszlik. (A széljegyzetben láthatsz olyan esetet, amikor a VeriSign modellje nem működik).
A bizalomháló legfőbb gyengesége abban lappang, hogy kevés felhasználó veszi a fáradtságot arra, hogy aláírja mások kulcsait, és ellenőrizze az aláírások valódiságát, azaz nem jön létre igazi háló. Az árva (aláírások nélküli) kulcsok jelensége sajnos teljesen szokványos. Természetesen a te kulcsaid nem maradnak aláírás nélkül, igaz? (Ugye, nem? Hiszen már írásunk puszta olvasásával is jó úton haladsz a PGP, illetve GnuPG bizalomháló erősítése felé.)
Ha már megemlítettem a PKI-kat, el kell mondanom, hogy léteznek PGP-kulcskiszolgálók is. Ezek hatalmas nyilvános kulcstárak, de csak a kényelmet szolgálják. A PKI-től eltérően, ahol a hitelesítő szervezettől letöltött kulcsban a meghatározás alapján megbízhatsz, a bizalomháló esetében nincs biztosíték az eredetre. A PKI-t az anyakönyvi bejegyzéshez hasonlíthatjuk, a PGP kiszolgálókat pedig egy klub havonta megjelenő hírleveléhez. Mindkét forrásból megtudhatod valakinek a születésnapját, de a két forrás megbízhatóságában nagy a különbség.

A GnuPG beszerzése, fordítása és telepítése

A GnuPG, mint már említettem, a legtöbb Linux-terjesztés része. Minden biztonsággal kapcsolatos programra, tehát a GnuPG-re is igaz, hogy nem árt, ha a legfrissebb változatot használod. Időnként érdemes ellátogatni a ➲ http://www.gnupg.org weblapra és tájékozódni az új kiadásokról.
Természetesen a GnuPG forráskódja is innen tölthető le, amire akkor van szükséged, ha a programot forrásból szeretnéd fordítani, esetleg nincs futtatható csomag a rendszeredhez. Egyszerűen töltsd le a csomagot egy tetszőleges könyvtárba (a /usr/src jó választás), és add ki a következő parancsokat:

```
cd /usr/src
tar -xzvf gnupg-1.0.6.tar.gz
cd gnupg-1.0.6
./configure
make
make check
make install
```

Megjegyezném, hogy a csomag neve eltérhet. Amikor ezt a cikket írom, a GnuPG legfrissebb változata az 1.0.6-os. Mikorra azonban olvasni fogod az írásomat, előfordulhat, hogy már kiadták egy frissebb változatát. Ugyanez vonatkozik a `tar` fájl kicsomagolása után létrejövő könyvtár nevére is.
Bár a `make check` parancsot nem szükséges kiadni – régebbi gépeken időigényes lehet –, én mégis hasznosnak tartom: részleteket ír ki a támogatott titkosítási algoritmusokról és próbaprogramok egész hadát futtatja le. Így azonnal kiderül, ha valami nem sikerült a GnuPG fordítása közben. Ha a próbaprogramok valamelyike hibával tér vissza, nincs értelme telepíteni a programot, előbb meg kell szüntetni a hiba okát.
Ha gondod támad a próbaprogramok futtatásakor, nézd meg az INSTALL és README fájlokat, esetleg megoldásokra bukkansz.

GPG rendszergazdai jogosultsággal?

Valószínűleg tudod, hogy a SUID (set user ID) bit használata általában kerülendő. A SUID bit – amelyet a `chmod` paranccsal állíthatsz be – hatására az adott program annak a felhasználónak a jogosultságaival fog futni, akinek a tulajdonában van, függetlenül attól, hogy ki indította el.
Ha például egy program nevének listázásakor (az `ls -l` parancs segítségével) a felhasználó jogosultságainál az `x` helyett `s` áll, és a program tulajdonosa a rendszergazda, ez azt jelenti, hogyha bármikor elindítjuk a programot, az rendszergazdai jogosultságokkal fog rendelkezni, vagyis ugyanazt megteheti, amit a rendszergazda.
Bizonyos esetekben egyes programok ok nélkül kapják meg a SETUID bitet, miközben tulajdonosuk a rendszergazda-felhasználó. A `gpg` nem ezek közé tartozik, érdemes SETUID bittel futtatni (SETUID=root, mert a `gpg` program a rendszergazda tulajdona), ami csökkenti annak az esélyét, hogy egy felhasználó titkos kulcsot vagy jelmondatot tudjon kiolvasni a memóriából. Ajánlom, hogy a `make install` után a következő parancsot add ki: `chmod u+s /usr/bin/gpg`.

GnuPG-gyorstalpaló: a digitális aláírás ellenőrzése

Miután telepítetted a `gpg`-t (forrásból a fent leírtak alapján, vagy Linux-terjesztésed CD-lemezéről), készen állsz arra, hogy létrehozd a saját kulcspárodat, és elkezdd felépíteni a bizalomháló rád eső részét. Mivel már kevés hely áll rendelkezésemre, egy olyan műveletet ismertetek, amihez nem kell saját kulcspár: kövessük végig valaki más aláírásának az ellenőrzését.
A digitális aláírás elkészítése elterjedt módszer (immár hazánkban is elfogadott – a szerk.), segítségével megbizonyosodhatunk róla, hogy a felhasználó által letöltött programcsomag megegyezik a fejlesztő által feltöltöttel.
A különálló aláírást (a PGP-aláírás lehet külön fájlban vagy csatolt fájlként ahhoz kapcsolva, amit aláírtak) a `gpgv` programmal ellenőrizzük. Ha a programnak megadunk egy aláírást, de nem rendelkezünk az aláíró nyilvános kulcsával, a `gpgv` hibát jelez. ***Listánkon*** egy ilyen próbálkozást láthatunk, nézzük meg közelebbről! Három parancsot adtunk ki:

```
gpgv gpa-0.4.1.tar.gz.sig
gpg --keyserver pgp.mit.edu \
       --recv-keys 621CC013
gpgv --keyring pubring.gpg \
       gpa-0.4.1.tar.gz.sig
```

A `gpgv` (amely az aláírások ellenőrzésére szolgáló egyszerűsített `gpg`) első futtatásánál csupán az ellenőrzendő aláírást adtam meg. Ha meglett volna a megfelelő nyilvános kulcs a `gpgv` nyilvános kulcscsomóján (***$HOME/.gnupg/trustedkeys.gpg***), ez a parancs lefutott volna, de mivel nem birtokoltam a szükséges kulcsot, hibával tért vissza.
A második alkalommal a `gpg`-t a `--recv-keys` kapcsolónak megadott kulcsazonosítóval futtattam. Ezt az azonosítót az előzőleg futott `gpgv` kimenetéből emeltem át. Meg kellett még adnom, hogy a `gpg` a pgp.mit.edu kulcskiszolgálón keresse a meghatározott kulcsot. A program megtalálta a kulcsot és sikeresen visszatért.
A harmadik parancsnál a `--keyring` kapcsolóval megadtam a `gpgv`-nek, hogy a nyilvános kulcsot az alapértelmezett kulcscsomón keresse a ***$HOME/.gnupg/pubring.gpg*** fájlban.
A parancs ezúttal megfelelő volt.
Egy valami maradt ki a fenti példából: természetesen a letöltött kulcs valódiságának ellenőrzése, azaz meg kell győződni róla, hogy a kulcs tényleg Werner Koch saját kulcsa. A feladat nem nehéz – körülbelül húsz másodpercembe került. Rákerestem a ➲ http://www.google.com-on a `621CC013 werner koch` karakterláncra.
A találatok között számos levelezési listára elküldött levél volt, amelyekben Werner aláírása – és benne a kulcs azonosítója – szerepelt. Ha valakinek sikerülne is arra a webkiszolgálóra betörni, ahonnan a programcsomagot letöltöttem, és elhelyezne ott egy trójai falovat vagy vírust tartalmazó fájlt, továbbá azt egy hamis kulccsal aláírná, a hamis nyilvános kulcsot pedig közzétenné a pgp.mit.edu kiszolgálón, nem érne el vele sokat, hiszen egy gyors webes kereséssel könnyen fény derülhet a csalásra. Kétlem, hogy akár a legelszántabb csaló is képes legyen arra, hogy a nyilvános kulcs minden közzétett példányát (weben, levelezőlista-archívumokban) felkutassa és megváltoztassa. Láthatjuk, hogy a bizalomháló működhet, feltéve, hogy óvatosak vagyunk, és alkalmanként szánunk egy kis időt az ellenőrzésre is.
E hónapra nincs már több helyem, de a következő részben a GnuPG számos további tulajdonságára is kitérek még. Remélem, sikerült felkeltenem az érdeklődést, a cikk folytatásáig pedig ajánlom, olvassák el az 58. oldalon található írásunkat ***A konzolos levelezés alapjai és a titkosítás*** címmel.

Mick Bauer (mick@visi.com)
hálózati biztonsággal foglalkozó szaktanácsadó. 1995 óta a Linux elkötelezett híve, 1997 óta pedig OpenBSD prófétaként tevékenykedik. Mick minden kérdést és megjegyzést szívesen fogad.

Kapcsolódó címek

The GNU Privacy Handbook
➲ http://www.gnupg.org/gph/en/manual.html nem annyira célirányos, mint a gpg(1) súgóoldala, írásunknál viszont részletesebb.

A GnuPG weblap ➲ http://www.gnupg.org a GnuPG legfrissebb változataira tartalmaz hivatkozásokat.

GPA és sok hasznos leírás
Az OpenPGP Alliance weblapja ➲ http://www.openpgp.org

PGP: Pretty Good Privacy *Simson Garfinkel* tollából az O'Reilly & Associates kiadásában, 1994. Annyiban ma már idejétmúlt, hogy még a GnuPG és az OpenPGP előtti időkből származik. Leírja, hogy mi a PGP, honnan származik és mire jó.

PGP Security ➲ http://www.pgp.com Phil Zimmermann által alapított cég, amit a NAI (Network Associates) felvásárolt. Phil már nem dolgozik a cégnél. A mai napig ez mind a kereskedelmi, mind a szabadon használható PGP program „hivatalos" forrása.

Phil Zimmermann weblapja
➲ http://www.philzimmermann.com

Az RSA-hoz intézett gyakran ismételt kérdések a titkosításról napjainkban, 4.1-es változat (más néven az RSA Crypto GYK) ➲ http://www.rsalabs.com/faq/index.html
Az egyik legjobb hálózaton megtalálható anyag a titkosításról, főleg mert nagy része a nyilvános kulcsú titkosításról szól.