Az asztali rendszerek biztonságáról (1. rész)

A számítógépek biztonsága napjaink egyik legfontosabb kérdésévé vált. UNIX alapú rendszereken leginkább a szerverek biztonsága a fő téma, mivel elterjedt vélekedés, hogy biztonságosabbak a Windows rendszereknél.

Ez általában igaz is, de az egyéni felhasználó végül könnyen óvatlanná válhat, az ismeretek hiánya vagy a nem kellő odafigyelés pedig akár helyrehozhatatlan károkat is okozhat. A desktop gép szinte állandóan ostrom alatt van: otthon gyermekünk, munkahelyünkön a munkatársak vagy éppen főnökünk tehet valami nem kellemes dolgot gépünkkel.

Nem beszélve a nyilvános helyeken lévőkről: könyvtárban, internet kávézóban, e-ponton működő desktop rendszerekről: ezek biztonsága még inkább fontos kérdés. A fenyegetés ezen kívül jöhet a belső hálózatról és az internetről egyaránt. De *Murphy* óta azt is tudjuk, hogy ami elromolhat, az el is romlik, tehát végső soron a számítógép önmaga ellenségévé is válhat. Ideje tehát megerősíteni saját vagy cégünk gépeit!

Mit is védünk?

Sokszor elégnek érezzük, ha munkahelyi rendszergazdánk felkészültnek mutatkozik és mindenféle trükkös módszerekkel körülbástyázza a szervert. Otthon pedig ha jelszóval kell bejelentkeznünk, máris „védve van a gép”. De mit is veszíthetünk?
Ezt érdemes átgondolnunk. Ha csupán internetezünk, akkor nem sokat, hiszen legfeljebb időveszteség ér bennünket az újratelepítés miatt. Sokan úgy gondolják, a nyílt forráskód világa, és a linuxos munkaállomások védve vannak a hálózatról érkező támadásokkal szemben.
Pedig ez nincs így. Valószínűleg fontos dolgokat is tárolunk gépünkön, amit nem szeretnénk más kezébe adni vagy éppen örökre elveszíteni. Nagy gonddal csiszolt beállításaink is képezhetnek értéket, amikkel kényelmesre és kellemesre faragtuk kedvenc operációs rendszerünket. Ha email klienst is használunk, akkor általában teljes levelezésünk is gépünkön tárolódik, amit nem szeretnénk mások orrára kötni, sőt az is bizalmas információ tárgyát képezi, ami a böngésző gyorsítótárában vagy az előzményekben megőrződik.
Nyilvános helyen lévő gépen általában erős korlátozásokat is be kell vezetni, hogy a felhasználók ne tehessenek meg bármit, sőt inkább csak egy-két, általunk kívánatosnak tartott dolgot (pl. böngészőhasználat, adatbázis elérése). Védeni kell a gépet az erőszakos újraindítások vagy feltörési kísérletek ellen is.
Meg kell akadályozni azt is, hogy nem megfelelő tartalmat töltsenek le, installáljanak fel, például egy trójait vagy egy rootkitet.
A felhasználói alkalmazások esetenként elég tág teret kínálnak a nem megfelelő használathoz, korlátozásaink tehát ide is kiterjedhetnek.
Az operációs rendszer szintjén is sok tennivalónk akad, mert ezek gyakorta csak kevés védelemmel településnek, kivéve az eleve biztonságosra tervezett rendszereket, mint amilyen az *OpenBSD*. A hálózati védelmünk gyakorta rendszergazdánk kezében van, otthon azonban – vagy ha mi vagyunk a hálózati rendszergazda – ennek megoldása is ránk hárul.
Védeni érdemes még fájlrendszerünket is, ha úgy véljük, egy esetleges támadó hozzáférhet adathordozóinkhoz, így azokat esetleg könnyedén lemásolhatná vagy módosíthatná. Az indítási folyamatba való esetleges beavatkozás is szóba jöhet. Ha letiltunk bizonyos eszközöket a *BIOS*-ban, úgy ennek védelme sem felesleges. Védelmünk legkülső eleme a hardver és annak elhelyezése, ami már fizikai védelem, ezért ezzel itt nem fogunk foglalkozni.
A biztonsági intézkedések tárgyalását abban a sorrendben tárgyalom, ahogy a gépünk is elindul, tehát a bekapcsolási folyamattól kezdve az operációs rendszeren keresztül a felhasználói alkalmazásokig és az általuk kezelt adatokig.

A BIOS védelme

Ez tulajdonképpen azt szolgálja, hogy illetéktelenek ne módosíthassák a *BIOS* beállításokat (*supervisor password*), illetve csak a jelszó ismeretében töltse be az operációs rendszer (*user password*). Mivel az egyes *BIOS* típusokhoz az interneten megkereshető általános jelszó is, amit a gyártó beleéget a chipbe, vagy pedig a gép szétszedése után a jelszó

törölhető, így ez elég gyenge védelem, viszont arra mindenképpen jó, hogy megakadályozza a kezdők károkozását, a profik számára pedig azt, hogy gyorsan behatoljanak a gépbe.
A *BIOS*-ban letilthatjuk a nem kívánatos külső perifériák használatát – floppy, *CD*/*DVD* meghajtó, sőt *USB* – , amiket arra használhatnak, hogy feltörjék a gépet. Természetesen ha rendszeresen használjuk ezeket a perifériákat, akkor ez nem túl kényelmes megoldás, hiszen minden egyes használat előtt vissza kell állítani ezeket a beállításokat, majd újra beállítani. Megoldás lehet még ezen eszközökről való bootolási lehetőségek tiltása is.
Persze a *BIOS* említett viszonylag egyszerű feltörhetősége miatt a nyilvános és jobban védett helyeken célszerű megoldás a fizikai kontaktust is megszüntetni, tehát kihúzni ezen eszközök adatkábelét, tápkábelét és csak akkor összeállítani, amikor esetleg installálunk valamit. Ez már elég komoly akadály lehet a próbálkozóknak, még a „varázsjelszóval" felszerelkezve is, a gép házának megbontásával remélhetőleg már kevesen élnek.

A bootmanager biztonsága

Egy *UNIX*-szerű rendszer használata esetén célszerű *GRUB*-ot vagy *LILO*-t telepíteni a boot szektorba, amelyek menürendszerrel láthatók el, így képesek többféle operációs rendszert, eltérő verziókat is elindítani. Itt kétféle védelemmel is élhetünk: megakadályozhatjuk, hogy illetéktelenek bármilyen operációs rendszert elindítsanak vagy pedig hogy nem biztonságos módon (például bejelentkezési shell nélküli) rendszert indíthassanak el. Az elsőnél jelszóval kell védenünk a boot managert, a másodiknál pedig nem szabad engednünk az automatikus bejelentkezést az operációs rendszer elindítása után.
Ez utóbbi történhet úgy is, hogy a bootmanager menüjéből eltávolítjuk a *„single"* módú indítási lehetőségeket vagy úgy is, hogy bár ezt engedjük, a *„single"* módban beállítjuk, hogy kötelezően be kelljen jelentkezni.
GRUB jelszó létrehozására kétféle lehetőségünk van: az egyiknél szimpla szöveges módban beírjuk a jelszót a konfigurációs állományba, a másiknál md5 titkosítást alkalmazunk (egy kis „sóval (salt) fűszerezve"). Az *md5*-ről és a *salt*-ról később részletesen fogok írni.
A jelszó titkosítás nélküli tárolásához a következőket írjuk a */boot/grub/menu.lst* fájl elejére:

```
timeout 10
password jelszavam
```

Az első paraméterrel megadjuk, hány másodpercet várakozzon, míg a `default` bejegyzéshez tartozó paraméterekkel elindítja a betöltést.
A második paraméternél a „jelszavam" helyére írjuk be saját jelszavunkat. Természetesen nem túl célszerű ennyire könnyen olvasható módon megadni egy jelszót, ezért titkosíthatjuk azt.
Ehhez adjuk ki a következő parancsot, ami bekéri a titkosítandó jelszót, majd kiírja az eredményt:

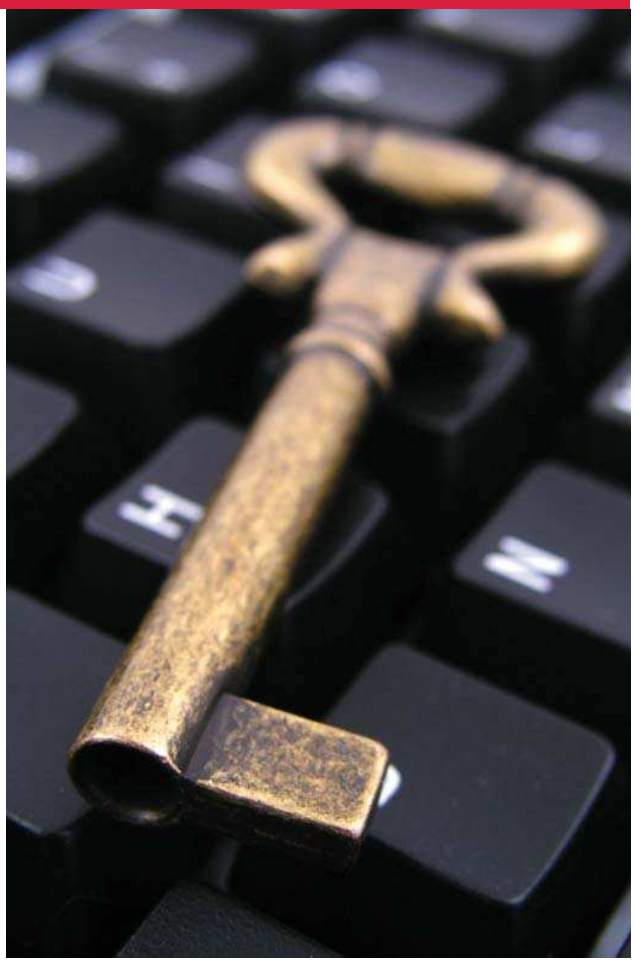
```
grub-md5-crypt
Password: *********
Retype Password: *********
Encrypted: $1$j9fu8/
➥Ha$5JM4n2wZOR.Zuo9iI0vJAS1
```

Másoljuk ki a titkosított karaktersorozatot. A */boot/grub/menu.lst*-be pedig illesszük be:

```
password --md5 $1$j9fu8/
➥Ha$5JM4n2wZOR.Zuo9iI0vJAS1
```

A védeni kívánt menübejegyzésekhez szúrjuk be a `lock` paramétert. A `grub-install` paranccsal pedig véglegesítsük a változásokat.
Legközelebbi indításkor a *GRUB* menüjében a védett elemnél a *P* billentyűt kell lenyomnunk, ekkor beírhatjuk a jelszót, ami feloldja a védelmet.
LILO-nál nem tárolhatjuk titkosítva a jelszót. Két lehetőségünk van: globális és menüpontonkénti védelem.
Globálisnál az */etc/lilo.conf* fájlban a következőket kell megadnunk:

```
password=jelszavam
restricted
delay=10
```



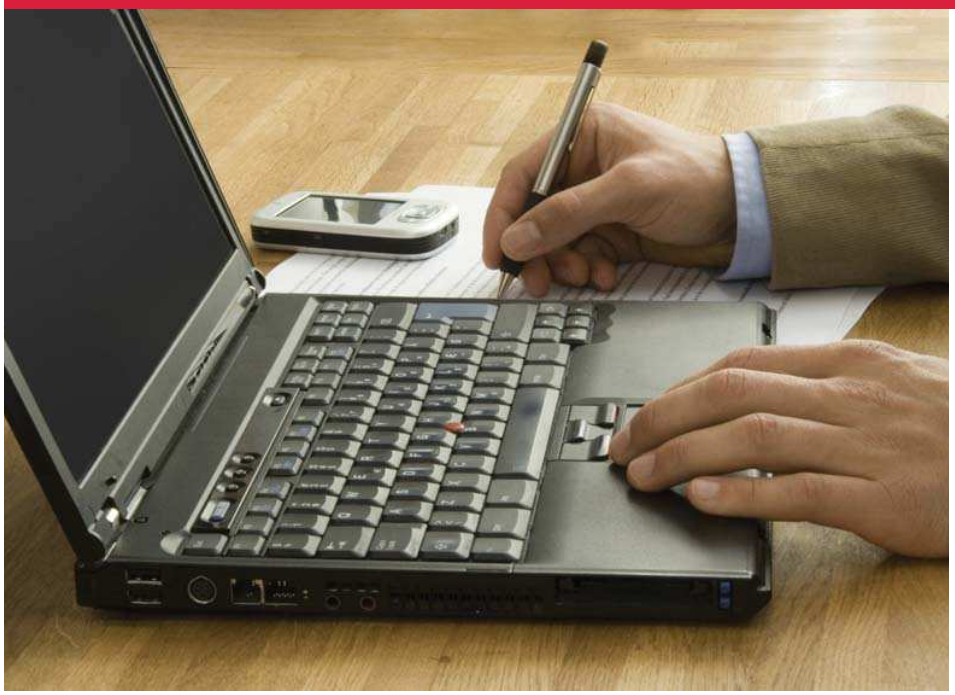
Minden egyes menüpontnál megadhatunk akár eltérő jelszavakat is:

```
image=/boot/kernel
        password=jelszavam
        restricted
```

A változtatás után ne felejtsük el kiadni a lilo parancsot.
A boot manager jelszóvédelme természetesen csak nehezítés támadónk számára. Ha például el tud indítani egy live Linux disztribúciót *CD*-ről, *DVD*-ről, floppyról vagy *USB* diszkről, akkor nem sokat ért ez a védelem, hiszen a boot managerre nincs is szüksége meghajtóink tartalmának eléréséhez. Viszont igen nehézzé tehetjük dolgát, ha ezeket nem képes végrehajtani, mert mondjuk egyik eszköz sem áll a rendelkezésére. Ez esetben már gyakorlatilag csak a merevlemezek ellopásával tudja adatainkat megszerezni.

Védelem a bejelentkezésnél

A bejelentkezés történhet szöveges terminálról, desktop menedzserből, távoli eléréssel, sőt akár



a soros portról is. Mindezeket megfelelően biztosítanunk kell, hogy ne érhessen bennünket meglepetés. Ha megnézzük az */etc/passwd* fájlt, valami ilyen fogad bennünket:

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/
➥bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/
➥sync
games:x:5:60:games:/usr/games:/
➥bin/sh
```

Az egyes mezők (kettősponttal elválasztva) a következők:

- bejelentkezési név
- jelszó titkosítva vagy x
- felhasználói azonosító
- csoport azonosító
- felhasználó neve vagy megjegyzés
- felhasználó alapkönyvtára
- felhasználó parancsértelmezője

Ha a jelszó helyén x-et találunk, akkor szerencsére *shadow*, azaz árnyékjelszavakat használunk, amik nem itt, hanem az */etc/shadow* fájlban találhatóak. Fontos, hogy ne itt tároljuk a jelszavakat, mert a *passwd* fájl mások számára is olvasható. Látható, hogy a legtöbb bejegyzés nem valódi felhasználót takar, hanem a rendszerfolyamatok számára kialakított, speciális felhasználói azonosítókat. Ezeknek nem szabad hogy jelszava legyen. De ajánlatos még a „root" felhasználó jelszavát is törölni a *shadow* fájlban:

```
root:*:13043:0:99999:7:::
daemon:*:13043:0:99999:7:::
bin:*:13043:0:99999:7:::
sys:*:13043:0:99999:7:::
sync:*:13043:0:99999:7:::
games:*:13043:0:99999:7:::
```

Ha a *shadow* fájlban csillag vagy felkiáltójel van a jelszó helyén, úgy azzal a felhasználói azonosítóval nem lehet belépni. A létező jelszavak természetesen titkosítva vannak:

```
felhasznalo:$1$O7T8tF83$V.kJdI5
➥P3iNPQjHWy6LpN.:13044:0:
➥99999:7:::
```

A speciális azonosítóknak is általában van alapértelmezett shellje, ami így furcsának tűnik, hiszen „ők" sohasem igényelnek ilyet. A biztonság kedvéért egyesek javasolják, hogy ezeket cseréljük ki `/bin/false`-ra, ami biztosítja, hogy ezen rendszerfolyamatok feltörése által sem lesz képes senki shellt szerezni.

A root jelszavának törlése pedig külön biztonságot ad arra, hogy a próbálkozók ne tudjanak teljes jogosultságot szerezni. Ilyenkor ne felejtsünk el a */etc/sudoers* fájlban saját magunknak minden jogot megadni:

```
felhasznalo  ALL=(ALL) ALL
```

Root jogokkal így a jövőben a `sudo` paranccsal futtathatunk valamit, root shellt pedig a

```
sudo su -
```

paranccsal szerezhetünk.
Amennyiben mégis jónak tartjuk, hogy a root-nak van jelszava, akkor érdemes szabályozni, hogy melyik terminálokról léphet be valaki root-ként. A */etc/securetty* fájlban szabályozhatjuk ezt, itt meg kell adni azoknak a tty terminálóknak a neveit (a /dev nélkül), amelyeknek engedélyezzük a root logint. Érdemes csak az első terminált megadni, a (`tty1`), a többit kikommentezni. Ha nem szeretnénk engedni a soros portról való root belépést, a `ttyS0`-t is kommentezzük ki, ha szerepelne benne:

```
#ttyS0
tty1
#tty2
#tty3
#tty4
...
```

Ezzel megakadályozzuk, hogy egyszerre több terminálon is be lehessen jelentkezni root-ként, így csak az első terminálra kell figyelnünk, ha otthagyjuk a gépet, hogy kijelentkezzünk a root shellből.
Ennél finomabban szabályozhatjuk a bejelentkezéseket a *PAM (Pluggable Authentication Modules)* segítségével. Az egyes alkalmazások autentikációjához különféle *pam* modulokat alkalmazhatunk. Ezek betöltését szabályozhatjuk a */etc/pam.d* könyvtárban lévő konfigurációs fájlokkal. Szöveges felületen, helyi terminálba a `login` alkalmazással tudunk bejelentkezni, ezért ehhez a *login* fájl kell szerkesztenünk. Keressük meg a következő sort az `/etc/pam.d/login` fájlban:

```
account required pam_access.so
```

és töröljük az elejéről a # jelet (vagy kézileg is beírhatjuk). A bejelentkezések szabályozása azután az */etc/security/access.conf* fájlban történik. Az egyes sorok formátuma a következő:

```
engedélyezés:felhasználók:
➥honnan
```

Az `engedélyezés` mező + vagy - lehet attól függően, hogy megadni vagy elvenni akarjuk-e a felhasználóktól, akik a „honnan” mezőben megadott helyről jelentkeznének be a bejelentkezés jogát. Használható az `ALL` kifejezés a mezőnevekben, ez az összes jogosultságot vagy az összes felhasználót, illetve az összes helyet jelentheti attól függően, melyik mezőben alkalmazzuk. Működik még az `EXCEPT` is, amivel kivételeket definiálhatunk. A `honnan` paraméter értéke lehet `LOCAL` is, ez a nem távoli elérést jelenti, azaz egy helyi konzolt.
Például tiltsunk meg minden helyi bejelentkezést a root kivételével az összes helyi terminálra:

```
-:ALL EXCEPT root:LOCAL
```

A távoli bejelentkezésekhez pedig csak a `csaba` nevű felhasználót engedjük meg:

```
-:ALL EXCEPT csaba:ALL EXCEPT
➥LOCAL
```

Egész bonyolult szabályrendszert is összeállíthatunk így.
Aki *SSH*-val jelentkezik be, az bizonyára észlelni fogja, hogy ezek a szabályok az *SSH*-ra nem érvényesek, ugyanis külön *SSH PAM* (*libpam_ssh* csomagban) modulunk van (*pam_ssh.so*). Ha az előbbihez hasonlóan akarjuk az ssh bejelentkezéseket is szabályozni a */etc/security/acces.conf*-ban, akkor (persze a *pam_ssh.so* modul megléte mellett) írjuk az alábbi sort a */etc/pam.d/ssh* fájlba:

```
auth required pam_access.so
```

És máris ugyanaz a szabályrendszer lesz érvényben az *SSH*-t használó felhasználókra is, mint a helyi bejelentkezésekre.

Az SSH-t saját konfigurációs fájljában is tudjuk némileg szabályozni (*/etc/ssh/sshd_config*). Nézzük az *sshd_config* fontos beállításait!
A `PermitRootLogin no` sorban megtilthatjuk (vagy engedélyezhetjük) hogy root-ként bárki *SSH*-n keresztül bejelentkezzen. Ajánlott ezt megtiltani. Természetesen a `PermitEmptyPasswords no` sornál is a tiltás az ésszerű: ne lehessen jelszó nélkül belépni.
A hálózati bejelentkezésekkel (és az *SSH*-val) később még részletesen foglalkozunk. Most térjünk vissza a helyi géphez és nézzük meg, milyen lehetőségünk adódik még a bejelentkezés korlátozására.
Az */etc/login.defs* fájlban adhatunk meg különböző paramétereket. Ezekből álljon itt néhány.

```
FAIL_DELAY   3
```

Ezzel szabályozzuk, hogy hány másodpercig várakozik a hibás bejelentkezés után. Nyugodtan emeljük meg ezt az értéket, a próbálkozóknak hadd menjen el a kedve a jelszók beírogatásától!

```
FAILLOG_ENAB yes
```


Természetesen engedélyezzük a hibás bejelentkezések naplózását (a */var/log/faillog* fájlba).

```
PASS_MAX_DAYS     90
PASS_MIN_DAYS     3
PASS_WARN_AGE     7
```

Ezekkel a jelszavak élettartamát szabályozhatjuk (`PASS_MAX_DAYS`: hány napig érvényesek, `PASS_MIN_DAYS`: hány napot engedélyezünk még a jelszó lejárta után, `PASS_WARN_AGE`: hány nappal jelezzen a jelszó lejárta előtt).
Természetesen a grafikus bejelentkezésnél – *gdm*, *kdm* – is van mit beállítanunk. *gdm*-nél a */etc/gdm/gdm.conf* fájlt kell szerkesztenünk.
Az `AutomaticLoginEnable` értéke feltétlen `false` legyen, mert egyébként aki bekapcsolja gépünket, mindjárt az `AutomaticLogin` változó által beállított felhasználóként léphet be.
A biztonsági beállítások a `[security]` szekcióban vannak elkülönítve. Nézzük meg a legfontosabbakat!

- `AllowRoot`: Ajánlott, hogy `false` legyen az értéke, hogy még véletlenül se léphessünk be rootként, így ha valaki ezután odaülne gépünkhöz, korlátlan jogosultsággal garázdálkodhasson.
- `AllowRemoteRoot`: `False`, hogy távolból se jelentkezhessen be senki root-ként.
- `AllowRemoteAutoLogin`: Természetesen távolból se lehessen automatikus login – `false`.
- `RelaxPermissions`: értéke 0, ha csak a tulajdonos, 1 ha a csoport tag is és 2, ha bárki írhatja a fájlokat a gdm által. Hacsak nincs rá egyéb okunk, ennek értéke legyen 0.
- `CheckDirOwner`: Az előző változóhoz kapcsolódik, ellenőrizze-e a home könyvtár tulajdonosát. Ha `true` az értéke, akkor ellenőriz. Személyes tapasztalatom az, hogy ha az előző változó értéke 0, már akkor sem indul el a grafikus felület, amikor megváltozik a home könyvtár tulajdonosa.

A bejelentkezés védelmében még nagyon sok lehetőséget említhetnék, de most csak a legalapvetőbb intézkedésekre jutott hely, így nem szóltam a különböző fizikai eszközökről sem, amelyek a biztonságos azonosítást szolgálják, valamint a *PAM* rendszerről is sokat lehetne még írni. Akinek ez kevés, a témáról nagyon sok és jó információt találhat az interneten.

A következő számban a fájlrendszer biztonságáról lesz szó.

Molnár Norbert
(molnar.norbert@gmail.com)

35 éves, rendszergazdaként dolgozik, 5 éve foglalkozik Linuxszal. Főként a szabad szoftverek és a számítógépes biztonság érdekli.

Budapesten él feleségével és 2 éves kisfiával. Hobbija a csillagászat és a filozófia – lehetőleg jó vörösbor mellett.