

Könnyű álmok (10. rész)

A PAM használata a gyakorlatban

Sorozatunk előző cikkében (*Linuxvilág*, október 46. oldal) áttekintettük a felhasználók azonosítása kapcsán felmerülő kérdéseket, és általánosságban beszéltünk a Linux PAM rendszeréről. Írásunk célja, hogy megismertessük a PAM fontosabb alkotórészeinek használatát, és tanácsokkal szolgáljunk a beállításukkal kapcsolatban.

A PAM-rendszer fontosabb alapmoduljai

A PAM-rendszer az alapvető modulokat önműködően telepíti. Az alábbiakban felsoroljuk a leggyakrabban használt modulokat és fontosabb szolgáltatásaikat.

Általános PAM-hibakeresés

A „debug” kapcsoló

A PAM-modulok hibáinak felderítésére a debug kapcsoló használható. Beállításának hatására az adott modul hiba esetén a syslog(3) rendszerhíváson keresztül ír a rendszernaplóba. Mivel minden modul rendelkezik ezzel a kapcsolóval, a továbbiakban nem tárgyaljuk.

A PAM-rendszer fontosabb alkotórészei

A „pam_unix” modul

A pam_unix a legfontosabb és leggyakrabban használt modul a Linux-változatokban. A Unix-rendszerek hagyományos azonosítási (authentication) és feljogosítási (authorization) eljárásait biztosítja. A rendszer szabványos hívásait használja, tehát a */etc/passwd* és a */etc/shadow* állományokkal dolgozik. Érdeemes megjegyezni, hogy a *pam_pwdb* modul is hasonló szolgáltatásokat nyújt, a felhasználókat azonban adatbázisban tartja. Nagy felhasználószámú rendszereken érdemes alkalmazni.

- **account**

Kapcsolói: debug; audit.

A felhasználói számla érvényességének ellenőrzését teszi lehetővé. A *shadow* állomány olyan mezőket tartalmaz (expire; last_change; max_change; min_change; warn_change), amelyek a felhasználó jelszavának kikényszerített cseréjét vagy a számla zárolását teszik lehetővé [1]. Vigyázat, ha a *shadow* állomány a fenti mezők valamelyikét nem tartalmazza, az ellenőrzés nem hajtódik végre!

- **auth**

Kapcsolói: debug; audit; use_first_pass; try_first_pass; nullok; nodelay.

A felhasználó jelszavas azonosítását tesz lehetővé. Amennyiben több jelszavas azonosítás is be van állítva (lásd később a pam_ldap modulnál), a try_first_pass érték használata célszerű. Ilyen esetben a rendszer a felhasználótól nem kérdezi meg újra a jelszavát, hanem az első modul által bekért jelszót használja. Ha azt szeretnénk, hogy a felhasználó több jelszóval lépjen be, ne használjuk. A nullok kapcsoló olyan felhasználók rendszerbe lépését teszi lehetővé, akiknek *shadow* állományában a kódolt jelszó mezője üres. A használata nem javasolt.

- **password**

Kapcsolói: debug; audit; nullok; not_set_pass;

use_authok; try_first_pass; use_first_pass; md5; bigcrypt; shadow; nis; min; max; obscure; remember.

A felhasználó szabványos jelszócseréjét teszi lehetővé. Az md5 és bigcrypt kapcsolók segítségével elérhető, hogy a jelszó ne a klasszikus crypt [2.] eljárással kódolva kerüljön a végleges helyére, hanem a megadottal. Ne felejtjük el beállítani, különben a rendszeren csak nyolc karakter hosszú jelszavakat lehet használni! Itt is alkalmazható a try_first_pass, ha a felhasználónak a különböző jelszótárakban egyforma jelszót szeretnénk beállítani. A use_authok beállítás a modul számára kötelezővé teszi az előző modul által átadott jelszó beállítását. Erre a pam_cracklib használata esetén van szükség (lásd később). A not_set_pass kapcsoló segítségével leltíthatjuk, hogy a bekért régi vagy új jelszó bármely más modulnak átadásra kerüljön. A nis kapcsoló hatására a rendszer a jelszó beállítására a NIS RPC-t használja. A min és max beállításával szabályozható a beállítható jelszó legkisebb és legnagyobb hossza. A min-t átlagos felhasználói rendszernél célszerű legalább nyolcra, erősen védett rendszernél pedig tízre állítani. Az obscure a beállítandó jelszón néhány alapvető ellenőrzést végez, amelyek a következők lehetnek: a jelszó nem hasonlíthat túlzottan az előzőhöz, nem lehet túl egyszerű (jelszóhossz, a használt karakterek típusa stb.), nem lehet az előző jelszó fordítottja vagy oda-vissza megegyező (például „qwertrewq”).

- **session**

Nincs kapcsolója.

Használatával a felhasználó neve és a szolgáltatás a munkamenet (session) elején naplózódik (a leírás szerint a végén is, de a tapasztalat ennek gyakran ellentmond).

A „pam_deny” és a „pam_nologin” modul

A deny segítségével megakadályozható a felhasználó adott szolgáltatáshoz való hozzáférése. Az auth és account esetén a felhasználó azonosítását és hozzáférést teszi sikertelenné, és amennyiben a password elemben használjuk őket, a felhasználó nem tudja megváltoztatni a jelszavát. A session alatt használva lehetővé teszi, hogy a felhasználó ne hozhasson létre munkamenetet.

A nologin modul a PAM-rendszer auth elemében elérhető, és a szabványos unixos nologin használatát teszi lehetővé. Amennyiben a */etc/nologin* állomány létezik, az azonosítás sikertelen. Leggyakrabban a rendszer indulásakor alkalmazzák, többfelhasználós rendszeren azonban kényelmes lehetőséget nyújt a felhasználók belépésének időleges tiltására egy esetleges karbantartás idején.

A „pam_securetty” és a „pam_shells” modul

A securetty és a shells modul meghatározza, hogy az adott felhasználó által használt terminál szerepel-e a */etc/securetty* állományban, illetve a felhasználó bejelnetkező héjja benne van-e a */etc/shells* állományban. Amennyiben az állomány az adott bejegyzést nem tartalmazza, a felhasználót



mindkettő elutasítja. Mindkét modul a PAM-rendszer auth eleméből érhető el.

A „pam_listfile” modul

A pam_listfile az azonosítási szakaszban egy állomány tartalmán keresztül teszi lehetővé a karbantartás engedélyezését vagy tiltását.

Lehetséges kapcsolói:

- onerr=succeed|fail
- sense=allow|deny
- file=állománynév
- item=user|tty|rhost|ruser|group|shell
- apply=user|@group

A modul veszi az item által meghatározott elemet (ahol a user a felhasználó neve; a tty annak a terminálnak a neve, ahonnan a kérés érkezett; az rhosts a távoli gép neve – ha van; az ruser a távoli felhasználó nevét adja meg – ha van; a group pedig a felhasználó csoportja), és megnézi, hogy a file által meghatározott állomány tartalmazza-e. Ha tartalmazza és a sense értéke allow, a modul sikerrel tér vissza, ha deny, akkor elutasító választ ad. Amennyiben hiba lép fel (például a meghatározott állomány nem létezik), az onerr által beállított értékkel tér vissza. Ezt érdemes fail-re állítani. Az apply kapcsolót akkor célszerű használni, ha a vizsgált elem terminál, távoli gép vagy héj. Segítségével a sikeres visszatérés egy felhasználóhoz vagy egy csoporthoz köthető.

Így tehát egyszerűen korlátozható egy adott szolgáltatás elérése. Használatára a legjellemzőbb példa az FTP, amelynél a listfile modult használták fel, hogy bizonyos felhasználók számára megtiltsák a szolgáltatás elérését. A beállító-állományban ez a következőképpen néz ki:

```
auth required pam_listfile.so item=user
  sense=deny file=/etc/ftpusers onerr=fail
```

Egyéb felhasználására is mutatunk példát a későbbiekben.

A „pam_limits” modul

Lehetőséget ad a felhasználók által használható erőforrások korlátozására, amire azért van szükség, mert a többfelhasználós Linux-kiszolgálókon nem engedhető meg, hogy egy felhasználó olyan mértékben terhelje le a rendszert, hogy a többiek (különösen a rendszergazdák) ne tudják a munkájukat zavartalanul végezni. A Linux-rendszer e korlátozások használatát sajnos csak kis mértékben támogatja, így a rosszindulatú felhasználóktól csak a hagyományos módszerek védenek meg tökéletesen (időleges vagy végleges kizárás, vasalt orru bakancs stb.).

Ezen keserű megállapítások a Linux 2.2.20-as és 2.4.14-es rendszermaggal folytatott hosszas kísérletezés után születtek. A felhasznált próbaprogramok az 1., 2. és 3. listán láthatók (24. CD Magazin/Konnyu könyvtár).

Amennyiben a felhasználók memóriafelhasználását korlátoztuk, a rendszer valamelyik fork-bombával túlterhelhetővé vált. Ha a belépésenkénti folyamatok (process) számát 4-re korlátoztuk, a fork-bombák akkor is szinte a teljes processzoridőt fel tudták használni, ráadásul az OpenSSH segítségével nem lehetett belépni a rendszerre. Tapasztalatunk szerint SSH-val csak akkor sikerült belépni a rendszerre, ha a lehetséges folyamatok száma legalább 40 volt. Mivel azonban a sikeres belépést követően a felhasználó 40 folyamatot futtathat, kedvezőtlen esetben a teljes processzoridőt le tudja foglalni. Az ésszerűtlen memóriafogyasztást meg lehet ugyan gátolni, de a sok memóriafoglalási kísérlet szintén megeszi a processzor idejének jelentős részét. Rendkívül kellemetlen, hogy ilyen esetben a legtöbbet a rendszermag dolgozik, így még korlátozni sem lehet.

A folyamatok számának korlátozása bizonyos esetekben a PAM-támogatás tökéletlen megvalósítása miatt nem megfelelő. Amennyiben a rendszeren a kifejezetten rosszindulatú felhasználókat ki tudjuk szűrni, van értelme a határok beállításának, mert ezzel csökkenthető a felhasználó akaratán kívül történő rendszertúlterhelés esélye. Sokat segíthet, ha a felhasználók nice szintjét csökkentjük, ezáltal hiba esetén a rendszergazdák nagyobb eséllyel tudnak sikeresen beavatkozni.

Eszményi az lenne, ha a felhasználóknak általános határokat lehetne beállítani (jelenleg csak a belépésenkénti létezik), és a rendszermag lehetővé tenné annak beállítását, hogy egy bizonyos felhasználó által kezdeményezett (felhasználó vagy rendszermag által végzett) feladat legfeljebb mekkora részt kaphasson a rendelkezésre álló processzoridőből (fair share scheduling). Amíg a hivatalos rendszermagban ezek nem valósulnak meg, addig a felhasználók korlátozása csak részleges lehet. Nem tartozik szorosan a témához, de itt érdemes megjegyezni, hogy a rendszermag lehetővé teszi annak a helynek a korlátozását, amit a felhasználók merevlemezen foglalhatnak. Így ésszerű mértékűre csökkenthető az egyes felhasználók terület-használata, és a levelesláda (mailbox) sem nőhet a többiek kárára egy adott méret fölé. Beállítása esetén azonban figyelni kell rá, mit tesz ilyen esetben a levelezőkiszolgáló.

Egyéb hasznos modulok

A pam_env modul (auth) környezeti változók előzetes beállítását vagy törlését teszi lehetővé. Többfelhasználós rendszeren célszerű alkalmazni, mivel a belépési héjtől függetlenül teszi lehetővé a környezet egységes beállítását.

A pam_rootok modul (auth) azonosítja a felhasználót, ha a felhasználói azonosítója 0. Ennek akkor lehet értelme, ha a rendszergazdát nem akarjuk egy szolgáltatás minden egyes használatakor azonosítani. A Linux-változatok legtöbbszörében a rendszergazdának megengedett a su használata jelszó nélkül, ami a beállítóállományban így fest:

```
auth sufficient pam_rootok.so
auth required pam_unix.so
```

A rendszergazda úgy tevékenykedhet bármelyik felhasználó nevében, hogy nem adja meg annak a jelszavát. Ez felvet bizonyos erkölcsi kérdéseket, ami azonban szinte minden rendszergazdai jogosítványnál felmerül. Különleges esetben egy finoman hangolt, külső szakértők által is felülvizsgált rendszernél elérhető a rendszergazdák jogainak a szükségesre történő csökkentése, de ez komoly hozzáértést és erőforrás-rafordítást igényel.

A pam_chroot modul (account, session, auth) segítségével lehetővé válik egy adott szolgáltatás root könyvtárának a PAM-rendszeren keresztüli beállítása. Hasznáról egy későbbi cikkben részletesebben írunk. A modullal jelenleg kissé nehézkes dolgozni, mivel a PAM-ot támogató programok egy része a munkamenetkezelést nem megfelelően valósítja meg. Jó példa erre az OpenSSH, ahol a PAM-megvalósítás félreérthetősége miatt a rendszer nem minden esetben működik helyesen. Többen kijavították az SSH hibáit, de a fejlesztők nem fogadták be a javításokat.

A pam_motd, pam_mail, pam_lastlog és pam_issue modulok a felhasználók tájékoztatását szolgálják. Belépéskor a terminálra a /etc/motd és a /etc/issue állományok tartalmát, az utolsó belépés idejét kiírják, továbbá jelzik, ha a felhasználónak új levele érkezett.

A pam_radius és pam_krb4 modulok segítségével a felhasználók azonosítása egy RADIUS- [3.] vagy Kerberos- [4.] kiszolgáló segítségével történik.

Ezek az azonosítási eljárások általában nagyobb hálózatokon használatosak, és nagy biztonságú azonosítást tesznek lehető-

vé. A pam_securetty segítségével egy állományban meghatározható, hogy mely terminálok tekinthetők biztonságosnak. A pam_time modul használata lehetővé teszi a hozzáférés idő szerinti korlátozását. Minden biztonsági rendszer alapvető eleme a megszokott és elfogadott engedélyezése, és a kirívó esetek tiltása. Amennyiben például valószínűtlen, hogy a rendszergazda reggel tíz óra előtt belépjen a konzolról, e modul segítségével letiltható, vagy egy PAM-ot támogató játékkalkulációval megoldható, hogy csak munkaidőn kívül lehessen elindítani.

A PAM különleges moduljai

A „pam_cracklib” modul

A unix modul password elemének kiegészítésére szolgál. Jóval finomabb jelszóbonyolultság-ellenőrzést tesz lehetővé. A unix modul obscure kapcsolónál említettekén kívül képes a szótári szavakon alapuló jelszavak kiszűrésére is. Kielégítő működéséhez egy megfelelő szavakat tartalmazó szótár szükséges, amit a legegyszerűbben oly módon állíthatunk elő, ha nagyobb mennyiségű levelezési listátartat szedünk össze, majd szavakra bontjuk. Célszerű olyan csomagokat is gyűjteni, amelyben a népek ékezzettel leveleznek, mivel a felhasználók előszeretettel tesznek ékezetes szavakat a jelszavukba. Továbbá célravezető összeszedni a felhasználók, valamint kisállataik és szeretteik adatait. A 4. listán látható (24. CD Magazin/Konnyu könyvtár) egyszerű kis Perl-programcska segítségével a szövegállományokat szavakká daraboljuk.

A program a bemenetén a tiszta szövegállományokat várja (kismértékben akár HTML-lapokat is, bár ettől leendő adatbázisunk feleslegesen hízik), és az adatok a kimenetén szavakra darabolva érkeznek. Az adatbázis a Debianon az alábbi parancsösszetétellel állítható elő:

```
cat sok sz vegÅllomÅny neve |
  ↘ mini_splitter | sort -u |
  crack_packer
  ↘ /var/cache/cracklib/cracklib_dict
```

Ezzel előállítottuk a szóadatbázist, amelyet a későbbiekben egyszerű rendszeresen frissíteni. A modul ellenőrzi, hogy a megadott jelszó nem képezhető-e valamelyik szótári szóból a kis- és nagybetűk valamilyen kombinációjával.

A modul a PAM password elemében működik, használata egyszerűsítve a következő: a felhasználó által megadott jelszó minden karaktere egy pontot ér, továbbá minden különböző karakterosztályba tartozó karakter egy jutalompontnyi számít. A rendszer számára meghatározhatjuk, hogy egy adott karakterosztályra legfeljebb mennyi jutalompontra adjon. Az ismert osztályok: kisbetű (lower), nagybetű (upper), számjegy (digit) és egyéb (other). A jutalompontok hangolása a következő értékek beállításával történik:

```
dcredit=N; ucredit=N; lcredit=N; ocredit=N,
Az N az adott osztály karaktereire adható legmagasabb plusz-
pontok száma. Amennyiben a jelszóban megadott szám alatti
vagy azzal megegyező számú adott osztályú karakter szerepel,
mindegyikükért egy pluszpont jár. A karakterszámból adódó
és a jutalompontok összegének legkisebbikét a minlen=N
értékkel szabályozhatjuk. Az N értéke a megengedhető
legkevesebb plusz egy. Így a következő beállításokkal:
dcredit=2 ucredit=1 lcredit=1 ocredit=2 minlen=12
csak olyan jelszó lesz elfogadható, amely vagy legkevesebb
10 kisbetűből áll, vagy ha van benne nagybetű, akkor nem rövi-
debb, mint 9 karakter; vagy ha van benne két számjegy és
nagybetű, akkor nem rövidebb, mint 7 karakter és így tovább.
A régi és új jelszó elvárt különbsége a difok=N értékkel állí-
tható be. A segítségével megadható, hogy egy adott jelszóban
```

hány karaktert kell mindenképpen lecserélni. Alapbeállítása tíz, de ehhez még egy újabb szabály adódik: amennyiben a jelszó karaktereinek legkevesebb a fele lecserélődik, felülbírálja az itteni beállítást, és a jelszó megfelel.

A „pam_ldap” modul

Nagyobb hálózatok estén gyakran felbukkanó gond, hogy a felhasználók a munkahelyek között vándorolnak, de mindenhol a megszokott munkakörnyezetet szeretnék látni. A rendszergazdák számára komoly nehézség lehet sok felhasználó együttes kezelése. Ilyen esetekben célszerű az LDAP-modul alkalmazása. A felhasználók adatait egy központi LDAP-kiszolgálón kell tárolni, így a tetszőleges munkaállomásra a megszokott jelszavakkal léphetnek be. A teljes támogatáshoz ne felejtjük el az nsswitch könyvtárakat sem áthangolni [5.]. A munkakönyvtárak átvitelére valamilyen hálózati állományrendszer is megfelel. Erre a célra jelenleg az NFS a legelterjedtebb megoldás, ami azonban biztonsági szempontból erősen megkérdőjelezhető, ezért használata kizárólag olyan környezetben fogadható el, ahol az ügyfelek tökéletesen megbízhatók – vagyis szinte sehol. Jelenleg a legésszerűbb a felhasználók munkakönyvtárait SSL-Sambán keresztül kijánlani, így lehetővé válik a biztonságos csatlakozás. A kis kitérő után térjünk vissza a központi felhasználóazonosításhoz.

Az LDAP-modul használatához először is szükségünk lesz LDAP-kiszolgálóra, amelyen a felhasználók adatait tároljuk. Mi az OpenLDAP 2.0.14-es változatát használtuk. A telepítés Debian Woody rendszeren a megszokott apt-get parancs segítségével egyszerű (a csomag neve *slapd*). Amennyiben a leendő LDAP szerkezetét előre megterveztük, telepítés közben létre lehet hozni a háttéradatbázist. A rendszer adatainak áttelepítésére tökéletesen alkalmas a PADL cég által fejlesztett MigrationTools nevű eszköz [6.]. Az OpenLDAP-nál az alapbeállítást kissé módosítani kellett, hogy a megfelelő sémameghatározásokat is betöltsék.

Ésszerű a hozzáférést is szabályozni, mert az alapterület bejelentkezés nélkül is olvasási jogot ad. A kissé paranoiásabb beállítás megfelelő része valahogy így fest:

```
access to attribute=userPassword
  by dn="cn=admin,o=Andrews,c=HU" write
  by anonymous auth
  by self write
  by * none
```

```
access to *
  by dn="cn=admin,o=Andrews,c=HU" write
  by self read
  by * none
```

Ezután megkezdődhet az LDAP-modul üzembeállítása, amely a */etc/ldap.conf* állományon keresztül zajlik. Lássuk az állomány tartalmát!

```
# Az LDAP-kiszolgáló neve
host tensor.andrews
```

```
# A keresős kiindulási pontjának DN-je
base ou=People,o=Andrews,c=HU
```

```
# A rendszergazda DN-je (a jelszava a
# /etc/ldap.secret állományban található,
# amely a libpam-ldap csomag telepítésekor
# kitöltésre kerül)
rootbinddn cn=root,o=Andrews,c=hu
```

```
# DN nevében keres a pam_ldap modul.
```

```
binddn cn=admin, o=Andrews, c=hu
bindpw ubertitkosjelszo
```

```
# A jelszavak t rol si form ja
pam_password md5
```

A PAM be llit allom nyaba kerul  sorokat p ld nkban adjuk meg.

Egy p ldarendszer be llit sai

P ld nk t rgya egy kisebb h l zat egyik felhaszn l i g pe legyen. A felhaszn l k gyakran v ndorolnak a g pek k z tt,  gy azonosításukat LDAP-on keresztül oldjuk meg. A felhaszn l k a rendszert a konzolr l val  bel p ssel  rik el, t volr l csak a rendszergazd k l phetnek be SSH seg ts g vel. Megmutatjuk a login  s az SSH PAM-modul be llit sait.

A login modul be llit sai csak annyiban t rnek el a megszokott l, hogy a felhaszn l kat LDAP-b l is lehet azonosítani. Amennyiben a felhaszn l  az LDAP-modul seg ts g vel azonosította magát, beengedj k, ha nem, megpr b ljuk a helyi felhaszn l i adatb zisb l azonosítani. Ezt eg szíti ki, hogy a pam_listfile modul haszn lat val bizonyos felhaszn l k rendszerr l val  id leges kitilt s t is lehet v  tessz k. Ennek kezelés re az 5. list ban (24. CD Magazin/Konnyu k nyvt r) tal lhat  egyszer  kis Perl-program szolg l. A seg ts g vel kilist zhatjuk a tiltott felhaszn l kat, felvehet nk  s t r lhet nk tilt st. Haszn lat nak megismer s hez haszn ljuk a -h kapcsol t.

P ldarendszer nk n a `/etc/pam.d/login`  gy n z ki:

```
# PAM be llit llom ny a 'login' szolg ltat shoz
```

```
auth requisite pam_listfile.so item=user
   sense=deny file=/etc/security/deny_users
   onerr=fail
auth requisite pam_securetty.so
auth required pam_nologin.so
auth required pam_env.so
auth sufficient pam_ldap.so
auth required pam_unix.so try_first_pass
```

```
account required pam_unix.so

session required pam_unix.so
session required pam_limits.so
session optional pam_lastlog.so
session optional pam_motd.so
session optional pam_mail.so standard noenv
```

```
password required pam_cracklib.so retry=3
  minlen=6 difok=3
password required pam_unix.so use_authok md5
password required pam_ldap.so try_first_pass*
```

Az SSH be llit sa a Debian  ltal feltelep tt l annyiban t r el, hogy fel lett v ve egy listfile modul, amely kiz r lag a `/etc/security/adm ns`  llom nyban felsorolt felhaszn l kat enged be. Az SSH PAM- llom nya:

```
# PAM be llit llom ny az 'ssh' szolg ltat shoz
auth requisite pam_listfile.so item=user
   sense=allow file=/etc/security/adm ns onerr=fail
auth required pam_nologin.so
auth required pam_env.so
auth sufficient pam_ldap.so
auth required pam_unix.so
```

```
account sufficient pam_ldap.so
account required pam_unix.so

session required pam_unix.so
session optional pam_lastlog.so
session optional pam_motd.so
session optional pam_mail.so standard
session required pam_limits.so
```

```
password required pam_cracklib.so retry=3
   minlen=6 difok=3
password required pam_unix.so use_authok md5
password sufficient pam_ldap.so try_first_pass
```

Amennyiben a felhaszn l  a rendszeren jelsz t v ltoztat, az LDAP-ban is meg kell v ltoztatni. Ehhez a passwd PAM-be llit sait a k vetkez k ppen kellett m dosítani:

```
# PAM be llit llom ny a 'passwd' szolg ltat shoz
```

```
password required pam_cracklib.so retry=3
   minlen=6 difok=3
password required pam_unix.so use_authok
   md5
password sufficient pam_ldap.so
   try_first_pass
```

Ezzel az alapbe llit sokat megv lasztottuk, a finomhangol st mindenkinek az  zl s re b zzuk. A r szletek  s mell khat sok tekintet ben n zz k meg a PAM-rendszer le r s t [7.]. Azon szolg ltat sokr l, amelyekr l hely hi ny ban b vebben nem tudtunk sz lni, elegend  adatot tal lunk a Linux PAM hivatalos honlapj n [8.], tov bb  számos egy b hasznos modulra is r akadhatunk. Mindenkinek hasznos keresg l st k v nunk!

Hivatkoz sjegyz k

- [1.] A *shadow*  llom ny form tuma - shadow (5)
- [2.] A *crypt* eljár s le r sa - crypt (3)
- [3.] RADIUS   http://www.gnu.org/software/radius/radius.html
- [4.] Kerberos   http://web.mit.edu/kerberos/www/
- [5.] Authenticating with LDAP using Openldap and PAM   http://www.imaginator.com/~simon/ldap/
- [6.] plain to ldap migration tools   ftp://ftp.padl.com/pub/MigrationTools.tar.gz
- [7.] A PAM-rendszer felhaszn l i k zik nyve   http://www.kernel.org/pub/linux/libs/pam/Linux-PAM-html/
- [8.] A Linux PAM-rendszer hivatalos honlapja   http://www.kernel.org/pub/linux/libs/pam/



M t  P ter (atya@andrews.hu), informatikus m rn k  s tan r. Biztons gi rendszerek ellen rz s vel  s telep t s vel, valamint oktatt ssal foglalkozik. 1995-ben tal lkozott el sz r linuxos rendszerrel. Ha teheti, kir ndul vagy olvas.



Borb ly Zolt n (bozo@andrews.hu), okleveles m rn k-informatikus. F k nt Linuxon fut  sz m t g pes biztons gi rendszerek tervez s vel  s fejleszt s vel foglalkozik. A 1.0.9-es rendszerem ideje  ta linuxozik. Szabadidej t bar taival t lti.