

```

{
    ...
    for (;;) {
        len = dlp_ReadNextModifiedRec(sd, db,
            ↪buffer, &id, &I, 0, &attr, 0);
        if (len < 0) break;
        if ((attr & dlpRecAttrDeleted) ||
            (attr & dlpRecAttrArchived))
            printf("T rløsre kijel lve: %ld", id);
    }
}

```

### Egy kicsi naplózás senkinek sem árt...

A Palm összehangolása után hasznos gyakorlat, hogyha a Palm naplófájljában megjegyzéseket is hagyunk róla. Akármint beleírhatunk, de az időpont és a dátum mindenképpen belekerül. Írjuk tehát az alábbi kódot programjaink végére:

```

{
    ...
    dlp_ResetSyncFlags(sd, db);
    dlp_CleanUpDatabase(sd, db);
    dlp_CloseDB(sd, db);

    dlp_AddSyncLogEntry(sd, "A Pilotr l
        ↪beolvastuk a m dos tÆsokat.\n");

    pi_close(sd);
}

```

### Mi kell az induláshoz?

Ha a Pilotot linuxos géppel szeretnénk használni, szerezzük be a *pilot-link* csomagot. A kezelőfelületek számos rendszerhez elérhetők (Next, BSD, Solaris, OS/2, Linux stb.). A segítségükkel Python, Java, Perl, Tcl, C/C++ nyelvű programokat írhatunk. A szükséges fájl a <http://ryeham.ee.ryerson.ca/pub/PalmOS> címen elérhető FTP-kiszolgálón található meg, a neve: *pilot-link.0.9.3.tar.gz*, linuxos gépen minden gond nélkül lefordítható. A csomag tényleg több, mint csak egy csatolókönyvtár a használatát bemutató egyszerű példaprogramokkal. Ezek az egyszerű eszközök nagyon hasznosak: a segítségükkel teljes biztonsági mentést készíthetünk a Pilotról (és vissza is állíthatjuk), adatokat és adatbázisokat másolhatunk róla és vihetünk föl rá stb. A könyvtár C/C++, Perl, Python, Tcl és még néhány más nyelvből hívható meg. Aki rendelkezik némi programozási készséggel, az a csomagba tartozó példaprogramok és cikkünk listái segítségével könnyűszerrel összeállíthatja a számára szükséges eszközöket.

Az új programokat, a leírás kiegészítéseit, a megjegyzéseket, a HOGYAN-okat az olvasók a [palm@duo.be](mailto:palm@duo.be) címre küldhetik, mi pedig a <http://www.duo.be/palm> című honlapon mindenki számára elérhetővé tesszük őket.

Telepítsük a csomagot az alábbi paranccsal:

```
tar -xzvf pilot-link.0.9.3.tar.gz
```

Ez létrehozza a *pilot-link.0.9.3* könyvtárat, benne a forráskóddal. Lépjünk is bele.

Adjuk ki a `./configure` parancsot, ami átnézi a rendszert a fordításhoz szükséges kiegészítő fájlokat keresve. A `configure` a programot alapértelmezés szerint a `/usr/local` könyvtárba telepíti. Ha ez nem megfelelő számunkra, a `./configure --prefix=K NYVT`R` paranccsal állítsuk be a telepítés helyét.

Adjuk ki a `make` parancsot, ami lefordítja a csomagot. Ekkor a fájlok még nem kerülnek a helyükre, így a végleges telepítés előtt alkalmunk nyílik kipróbálni a programot. Ha egy régebbi változatra egy újabbat telepítünk, ellenőrizzük, hogy minden működik-e. Általában természetesen semmiféle gond nem szokott adódni.

Rendszergazdaként adjuk ki a `make install` parancsot, mely a fájlokat a megfelelő könyvtárba helyezi. Ha rendszergazdaként nem tudunk belépni, akkor olyan helyre telepítsük a programot, amelyhez rendelkezünk írási jogosultsággal. Ne felejtjük el a futtatható fájlokat tartalmazó könyvtárakat beírni a rendszer alapértelmezett útvonalába (`PATH` változó). Nézzünk meg a csomag mellé kapott hasznos példaprogramokat is, amelyek leírását a *További érdekességek* részben említett helyen olvashatjuk el.



Johan Coppieters

([palm@duo.be](mailto:palm@duo.be)) a Duo nv nevű cég vezetője. A belgiumi Brugesben székelő vállalat Belgium legnagyobb vállalatai számára készít weboldalakat, belső hálózatokat és internetes alkalmazásokat.

Kevin Velghe

írta a C nyelvű Palm-Linux összehangoló programot, s egy három hónapos iskolai gyakorlat cseretanfolyam közben a kutatás egy részét is ő végezte. A Duo nv-nél találhatjuk meg.



## 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|rhosts|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. Célzerű olyan csomagokat is gyűjteni, amelyben a népek ékezzel 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 célzerű 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 felhasználó 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 kiindul 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.
```