Az LDAP szerver és kliens oldali vizsgálata

Diplomamunka

DE, IK

Készítette: Molnár Ildikó

Témavezető: Mecsei Zoltán

2007

Köszönetnyilvánítás

Köszönettel tartozom mindazoknak, akik diplomamunkám elkészítésében segítettek, és akik türelmükkel, támogatásukkal könnyítették meg munkámat.

A teljesség igénye nélkül külön köszönet illeti témavezetőmet, Mecsei Zoltánt hasznos és nélkülözhetetlen tanácsaiért.

Szintén köszönettel tartozom férjemnek, Rédai Zoltánnak, tapasztalatain alapuló ötleteiért, és végtelen türelméért.

Tartalomjegyzék

Bevezetés

Diplomamunkám témájául egy LDAP szerver és kliens gép beállítását választottam. Választásom azért esett erre a témára, mert az egyre növekvő régi és új alkalmazások megjelenésével, az ember már belefárad a hozzájuk tartozó különböző adatbázisok fenntartásába, karbantartásába. De vajon hogyan lehetne ezt orvosolni?

Nos, számos kollégával konzultáltam ebben a témában, de sajnos eleinte nem kaptam kielégítő választ. Majd egyszer valaki megjegyezte, hogy ez a probléma az LDAP-val könnyen kezelhető. Szégyenkezve vettem tudomásul, hogy igazából nem tudom mire gondol. Mi az az LDAP? Némi kutató munka után, konstatáltam, hogy valóban ez az ami nekem kell! Lelkesen hirdettem szerte a világba, hogy megtaláltam, ha nem is az élet értelmét, de az egyik leghasznosabb dolgot..

Persze, ami akkora számomra már érthetővé vált, embertársaim számára, csak négy betű érthetetlen sorozatát képezte. Eszembe jutott, hogy annak idején én is ilyen rémült arccal álltam a témával szemben.

Diplomamunkám célja az LDAP megismertetése, és a kezdeti lépések, konfigurációk bemutatása. Hiszen mint tudjuk, hálózati környezetben rendkívül nagy jelentőségű a fontos adatok szervezettségének fenntartása és gyors elérése. Ideális esetben egy központi szerver tárolja az adatokat egy címtárban és osztja szét a klienseknek egy meghatározott protokoll segítségével. Az adatok úgy vannak szervezve, hogy az alkalmazások széles skálája számára elérhetőek legyenek. Nem kell tehát minden egyes naptárprogramhoz és e-mail klienshez külön adatbázist fenntartani – elegendő egyetlen, jól karbantartott központi adattárat használni. Így sokkal kisebb fáradsággal, sokkal pontosabban karbantarthatók az adatok. Az LDAP nyílt, szabványos protokoll segít abban, hogy a lehető legtöbb kliensalkalmazás képes legyen az adatok elérésére.

Kezdetek

Abban a pillanatban, hogy egy IT infrastruktúrában egynél több rendszerkomponens (számítógép, hálózati eszköz stb.) jelenik meg, melyeken felhasználói azonosítást kell végezni, azonnal felmerül az igény, hogy ezt a feladatot központosított módon oldjuk meg. A központi nyilvántartásnak mindenképpen valamilyen adatbázisra kell épülnie, mely tartalmazni fogja a felhasználók természetes adatait (név, cím, telefonszám stb.), a biztonságos azonosításhoz szükséges adatokat (jelszó, publikus kulcs stb.) ill. néhány alkalmazás-specifikus paramétert. Vajon melyik a legalkalmasabb megoldás erre a feladatra?

1984-ben az ITU és az ISO közösen kezdett dolgozni egy szabványtervezeten, melynek célja, hogy a fent bemutatott adatok tárolására, visszakeresésére alkalmas biztonságos és elosztott megoldást adjon. Ez a szabvány az X.500 nevet viseli és 1998-ban jelent meg az első változata. Maga az X.500 egy áttekintést tartalmaz és a koncepciót fogalmazza meg, a további X.5xx szabványok (X.501, X.509, X.511, X.518, X.519, X.520, X.521, X.525, X.530) foglalkoznak a részletkérdésekkel. Ezek közül talán az X.509 azonosítási keretrendszerrel kapcsolatos szabvány a legismertebb, hiszen a jelenleg használatos publikus kulcsú azonosítási rendszerek az ennek a szabványnak megfelelő tanúsítványokat használják (SSL, TLS stb.). A teljes szabványgyűjtemény 1997-ben megjelent hármas változata (X.500v3) terjedt el szélesebb körben, de 2001-ben is került bele egy kiegészítés a negyedik változatban.

Maga az X.500 szabvány egy hierarchikus, objektum orientált adatbázist definiál névtárnak (directory). Ilyen típusú adatbázisokkal nap, mint nap találkozhatunk: állomány rendszer, DNS stb. Az adatbázis szerkezete egy kliens-szerver modellt követ. Az adatbázis szervereket névtárszolgáltatás ügynököknek (Directory Service Agent - DSA) nevezzük, a klienseket pedig névtár felhasználó ügynöknek (Directory User Agent - DUA). A kliens és a szerver egymással a névtár hozzáférési protokollon (Directory Access Protocol - DAP) keresztül kommunikál. Maga a DAP a teljes OSI hét rétegű adatmodellt definiálja és használja.

A teljes X.500 szabványrendszer megvalósítására is léteznek rendszerek és számos esetben használatuk indokolható is. Az 1990-es évek elején azonban a Michigan egyetem egy csapata úgy döntött, hogy a teljes X.500 DAP protokoll helyett, annak egy egyszerűsített változatát valósítja meg, ami az akkor már létező és nagy népszerűségnek örvendő TCP/IP pro-

tokollra épül az OSI helyett. E mellett még néhány további egyszerűsítést is végrehajtottak a protokoll adatmodelljében és végül 1993-ra így született meg a 1487-es számú RFC, ami a LDAP azaz a Lightweight Directory Access Protocol első változatát írta le. Ezt egy egész sor RFC követte, ami az LDAP-val kapcsolatos részletkérdéseket igyekezett tisztázni. Az így kialakult szabvány:

- meghatározza az adatok hozzáférésére használható hálózati protokollt,
- meghatározza a tárolandó információ szerkezetét és annak ASCII reprezentációját (LDIF),
- megmondja, milyen elnevezési konvenciót használhatunk az adatelemek megnevezésére (distinguished name - DN),
- definiál egy biztonságos azonosítási mechanizmust (binding),
- meghatározza az adatok elosztásának módját és az adatokon végezhető elosztott műveleteket,
- megmondja, hogyan lehet az információs modellt és a hálózati protokollt kiterjesztésekkel ellátni,
- kvázi szabványos C és JAVA API-t határoz meg

LDAP információs modell

Az LDAP protokollt beszélő szerveralkalmazások mindegyike ugyanúgy kell, hogy reprezentálja az általa tárolt információt, függetlenül attól, hogyan tárolja azt. Ennek az információs modellnek az alapja a Névtár Információs Fa (Directory Information Tree - DIT). Ebben a fában minden csomópontnak 0 vagy több gyermek csomópontja lehet és a gyökér csomópontot leszámítva minden csomópontnak, pontosan egy szülője van. Az információ a fa csomópontjaiban tárolódik.

A DIT minden csomópontjának adott hierarchia szinten egyedi nevet adunk (Relative Distinguished Name - RDN). A teljes DIT-ben az egyes csomópontok helyét egyértelműen megadja a gyökértől az adott csomópontig vezető úton vett csomópontok RDN-jeinek összessége (ez maga a Distinguished Name azaz a DN). A DIT szerkezetéből adódóan a névtárban nincs két elem, aminek azonos lenne a DN-je. A DN formátumát az X.521 ASN.1 formátumban határozza meg az RFC2253**,** azonban ezt jelentősen egyszerűsíti egy UTF-8 karakterláncra. E szerint az egyszerűsített szabvány szerint az RDN-ek típus=értek alakú párok; míg a

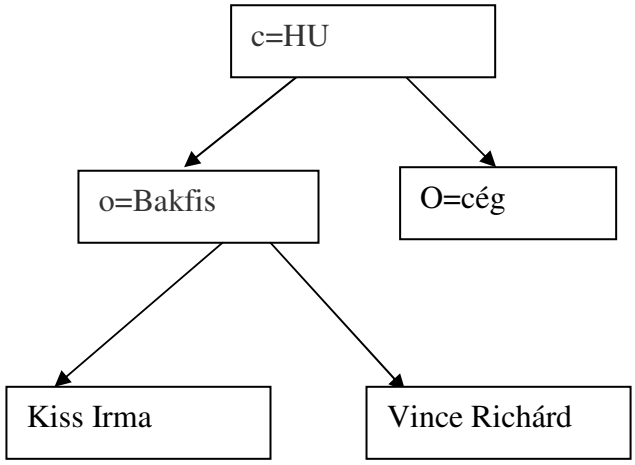
teljes DN ilyen RDN-ek vesszővel elválasztott sorozata. A lehetséges típusazonosítókat a LDAP séma határozza meg. Az alábbi táblázatban néhány példát foglaltunk össze.

Típus	**Attribútum szintaxis megnevezés**
CN	commonName - általános név
O	organizationName - szervezet neve
OU	organizationalUnitName - szervezeti egység neve
C	countryName – ország azonosító (pl.: hu, us)
DC	domainComponent - domain név elem (pl: dc=www,dc=bakfis,dc=hu)
UID	userid - felhasználó azonosító

RDN: c=HU

RDN: o=Bakfis

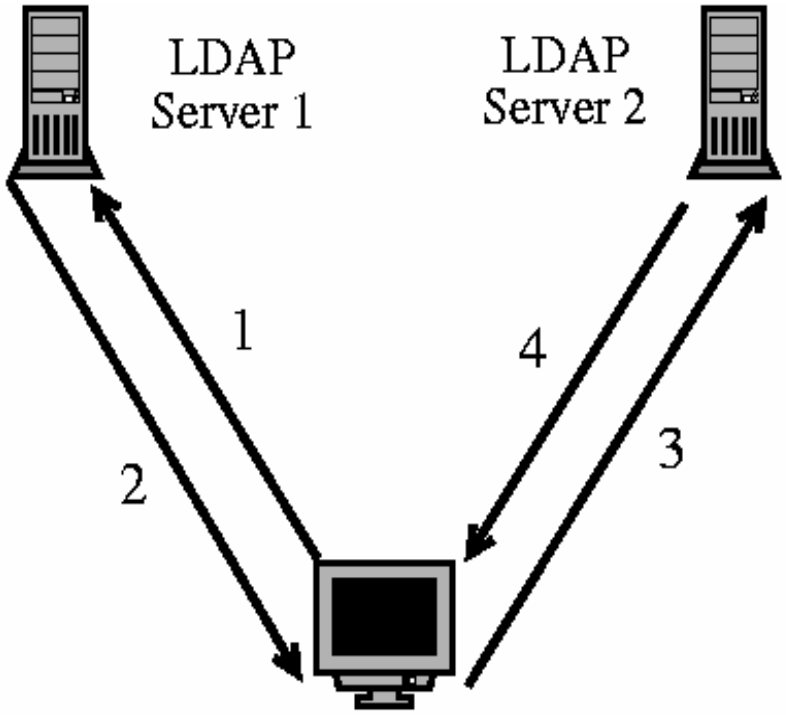
RDN: cn=Kiss Irma



DN: cn=Kiss Irma, o=Bakfis, c=HU

Az LDAP információs modell lehetőséget ad arra, hogy a DIT egyes alfái más-más DSA-kon legyenek, ezzel valósítva meg a DIT elosztottságát. Habár az X.500 modell szerint a világon egy egységes DIT létezik, aminek egyes alfái vannak csak elszórva és egy alfa csak egyszer használható, az LDAP-hoz kapcsolódó információs modell már ezt nem követeli meg, ilyen szigorúan.

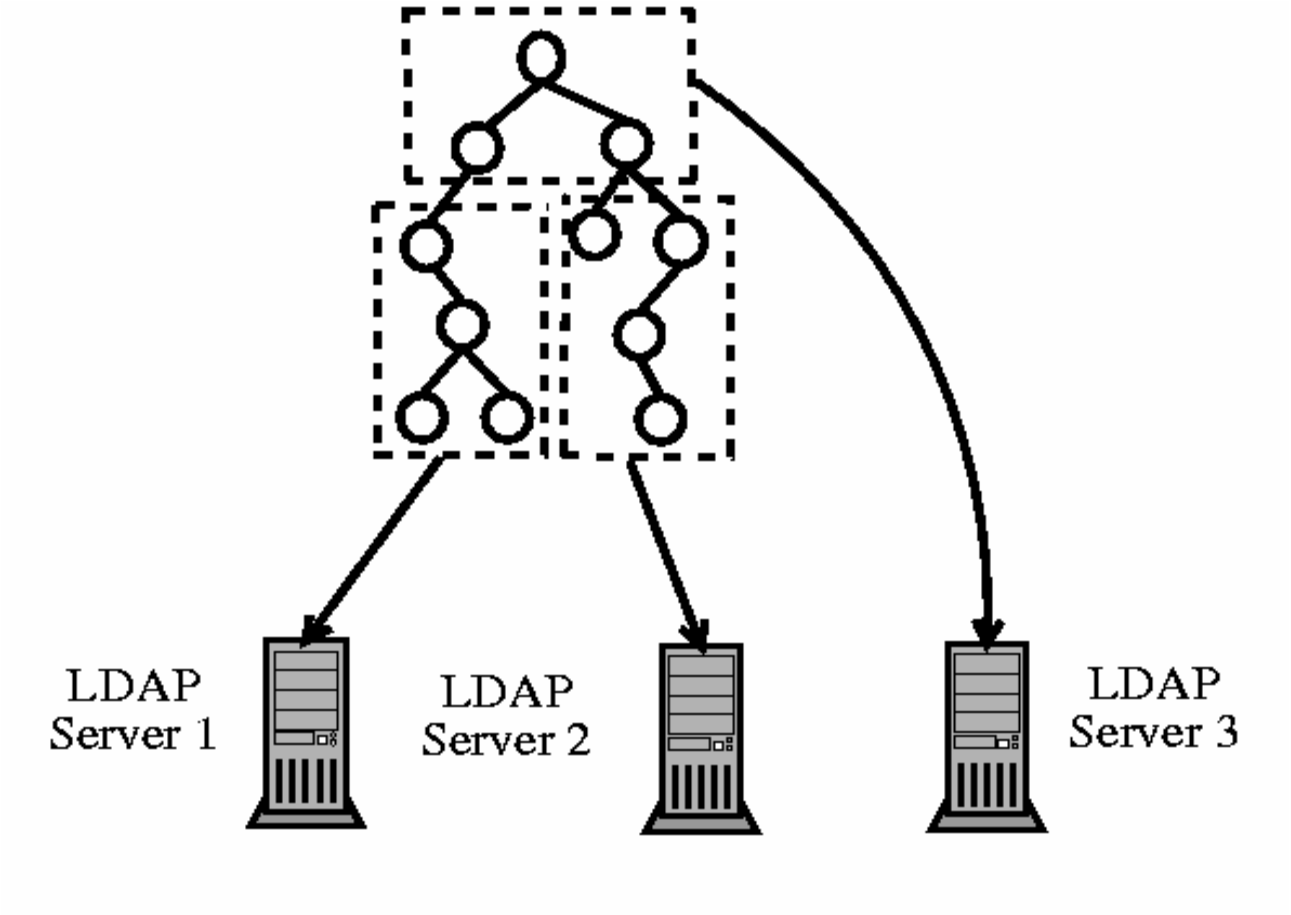
Hogyan működik az LDAP



1. A kliens információt kér
2. Server 1 visszaküldi a server 2 elérhetőségét
3. A kliens tovább küldi a kérdést server 2 -nek
4. Server 2 visszaküldi a kért információt a kliensnek

Az LDAP szolgáltatás kliens-szerver modellen alapul. Egy vagy több LDAP szerveren tárolt adatból épül fel az LDAP fa vagy LDAP háttér adatbázis. Az LDAP kliens egy LDAP szerverhez csatlakozik, és felteszi a kérdéseit. A szerver a válasszal vagy egy mutatóval reagál, hogy hol talál több információt a kliens (tipikusan egy másik LDAP szerver címével). Mindegy, hogy a kliens melyik LDAP szerverhez csatlakozik, ugyanazt a címtárat látja, ugyanaz a név az egyik címtár szerveren, ugyanazt az adatot jeleníti meg, mint a másikon. Ez fontos tulajdonsága az olyan globális címtárszolgáltatásoknak, mint az LDAP.

A rendszer globális nézetben



A hozzáférés alapja:

- Az attribútum gyűjtemény
- Distinguished name (DN) - mint domain name

De mi a címtár szolgáltatás?

A címtár olyan, mint egy adatbázis, de arra törekszik, hogy magába foglaljon egy részletesebb, tulajdonság alapú információkezelést. A címtárban az információt általában gyakrabban olvassák, mint írják. Ennek következtében a címtárak általában nem alkalmaznak bonyolult tranzakciókezelést vagy roll-back rendszereket, melyeket általában az

adatbáziskezelők használnak a nagyméretű, összetett frissítésekhez. A címtár-aktualizálás tipikusan egyszerű, mindent vagy semmit jellegű változás.
A címtárakat összetett kérdések gyors megválaszolására hangolták. Képesek arra, hogy széles körben replikálják az információkat, hogy növeljék az elérhetőséget és a rendelkezésre állást, miközben csökkentik a válaszidőt. A többszörözött címtár információk egyes példányai között átmeneti rendezetlenség megengedett, de a replikák végül szinkronba kerülnek.
Címtár szolgáltatás nyújtására számos lehetőség van. Különböző eljárásokkal más és más információk tárolhatók a címtárban, különböző követelmények szerint lehet hivatkozni az információkra, lekérdezhetők és frissíthetők, védhetők a meg nem engedett hozzáféréstől, stb. Néhány helyi címtárszolgáltatás korlátozott környezetben nyújt szolgáltatásokat (pl a finger szolgáltatás önálló gépen). Más szolgáltatások globálisak, széles körben elérhetőek.

Címtár alatt egy gyors, kifejezetten a hatékony olvasásra és keresésre optimalizált adatbázistípust értünk:

- A sok (egyidejű) olvasási hozzáférés érdekében az írási hozzáférés csekély számú frissítésre van korlátozva. Emlékeztetőül: a hagyományos adatbázisokat arra optimalizálják, hogy rövid idő alatt a lehető legnagyobb adatmennyiséget legyenek képesek fogadni.
- Az írási hozzáférés korlátozása azért nem számít komoly megszorításnak, mert a címtárszolgáltatásokat többnyire ritkán változó, statikus jellegű adatok tárolására és kiszolgálására használják. A hagyományos adatbázisokban tárolt adatok általában jóval gyakrabban változnak (*dinamikus* adatok). A vállalati címjegyzék telefonszámai sokkal ritkábban változnak, mint például a könyvelés adatai.
- Statikus adatok karbantartásakor ritkán kell frissíteni a meglévő adatokat. Dinamikus adatok használatakor – különösen, ha például bankszámlákról vagy könyvelésről van szó – az adatok konzisztenciája az elsődleges szempont. Egy pénzügyi műveletnél például nem elég, hogy az egyik fél számláját ugyanannyival növeljük, mint amennyivel a másikét csökkentjük: a műveleteket egyidejűleg kell végrehajtani egy *tranzakción* belül, hogy még a rendszer esetleges meghibásodása esetén se boruljon fel az elszámolás (és senki ne legyen megkárosítva). Az adatbázisok egyik legfontosabb funkciója az ilyen tranzakciók támogatása, míg a címtáraknál nem a tranzakciókon van a fő

hangsúly. A címtárak szokásos felhasználási módjaiban nem jelent problémát az adatok rövid ideig tartó inkonzisztenciája.

Hálózati környezetben rendkívül nagy jelentőségű a fontos adatok szervezettségének fenntartása és gyors elérése. Ebben segít ez a címtárszolgáltatás, amely – például a szaknévsorhoz hasonlóan – az adatokat jól szervezett, gyorsan kereshető formában teszi elérhetővé. Ideális esetben egy központi szerver tárolja az adatokat egy címtárban és osztja szét a klienseknek egy meghatározott protokoll segítségével. Az adatok úgy vannak szervezve, hogy az alkalmazások széles skálája számára elérhetők legyenek. Nem kell tehát minden egyes naptárprogramhoz és e-mail klienshez külön adatbázist fenntartani – elegendő egyetlen, jól karbantartott központi adattárat használni. Így kevesebb fáradsággal és sokkal pontosabban tarthatók karban az adatok. Az LDAP-hoz hasonló nyílt, szabványos protokollok használata segít abban, hogy a lehető legtöbb kliensalkalmazás képes legyen az adatok elérésére.

LDAP séma

A DIT csomópontjaiban található információt attribútum halmazok írják le, absztrakt módon. Egy attribútum egy névazonosítót és egy vagy több érteket tartalmaz. Az attribútumnak van típusa (string, numeric stb.), mely meghatározza a lehetséges értékek formai követelményeit (szintaxis) valamint egy összehasonlítási szabálya, mely szerint az értékek azonosnak tekinthetők (pl.: caseIgnoreMatch - kis és nagybetű különbségére érzéketlen).

Egy LDAP osztály meghatározza, hogy egy őt reprezentáló adatnak mik a kötelező és a lehetséges attribútumai, valamint definiálja ezeket az előbb bemutatott jellemzőkkel. Az osztályok között egyszerű egyszeres öröklődés valósítható meg. Az LDAP séma nem más, mint a rendszer által használható LDAP osztályok definíciója.

Az LDAP szabvány előírja, hogy minden attribútum és osztály névnek globálisan egyedinek kell lennie ezért minden sémaelemet (attribútum definíciót és osztályt) globálisan egyedi azonosítóval látnak el. Ez az azonosító az OID (Object Identifier). Az OID-ek hierarchikusan felépülő számhalmazok (pl.: Internet OID-ja: 1.3.6.1), melyek formátumát az X.208 ITU szabvány definiálja. Ha valaki hiányol bizonyos attribútumokat, akkor séma kiegészítéssel élhet, de ehhez először illik OID szám tartományt igényelni az IANA (Internet Assigned Numbers Authority) szervezetnél. Igyekezzünk ellenállni annak a csábításnak, hogy saját OID

tartományt igényeljünk, és csak akkor éljünk ezzel a megoldással, ha tényleg nem találunk más megoldást. Tesztelésre és helyi rövid idejű felhasználásra az 1.1.x alakú OID-eket vehetjük igénybe. Az OID-ek nem csak az LDAP sémánál, hanem az SNMP objektumok azonosítására is használják.
Az osztályokat megvalósító adatoknak speciális attribútumokkal kell rendelkezniük, mint például:

- *Objectclass:* megadja, mely objektumokat valósítja meg az adott adat,
- az adott hierarchia-szinten érvényes *RDN*-t mindenképpen tartalmaznia kell (pl.: ha a DN cn=Szalai Ferenc, o=Gluon, c=hu, akkor cn attribútumot mindenképpen definiálni kell)
- az LDAP szerver maga még bejegyezhet néhány speciális belső attribútumot többek között a bejegyzés készítésének időpontját, készítőjét, utolsó módosításának idejét stb.

Az LDAP séma készítésekor és a megfelelő objektumok kiválasztásakor figyelembe kell vennünk, hogy az objektumoknak is van típusa:

- **Abstract**: az öröklődés alaposztályai tartoznak ebbe a csoportba. Az adatok az ilyen osztályokat nem valósítják meg közvetlenül. A legtipikusabb példája a top objectclass, ami minden osztály ősosztálya.
- **Structural**: minden adatnak pontosan egy ilyen típusú ilyen osztályt valósíthat meg. Az előző szabály alkalmazásánál figyelembe kell venni az öröklődési függőséget is. Például, ha egy A structural objectclass szülője B structural objectclass-nak akkor az adat megvalósíthatja A-t és B-t is. Az ilyen típusú objectclass-ok írják le a legalapvetőbb dolgokat, mint személyek (Person), intézmények (Organization) stb.
- **Auxiliary**: kiegészítő attribútumokat határoznak meg és egy adat tetszőleges számú ilyen típusú objectclass-t megvalósíthat.

Az LDAP séma ASCII reprezentálására született ajánlás leírására a RFC2252-ben található.

LDAP adatok ASCII reprezentációja – LDIF

Számos esetben hasznos lehet az LDAP adatok szöveges formátumú platform független reprezentációjára. Például különféle LDAP szerverek közötti adatmigráció az egyik tipikus felhasználás, de igen hasznos lehet, ha az LDAP szerver adatait emberi fogyasztásra szánjuk. Erre találták ki a LDAP Data Interchange Formatot (LDIF).
Egy konkrét példa egy LDIF bejegyzésre az alábbi:

dn: cn=Kiss Irma, ou=Bakfis, c=hu
objectclass: top
objectclass: person
objectclass: posixaccount
cn: Kiss Irma
sn: Kiss
mail: kissir@bakfis.hu
loginShell: /bin/bash
uidNumber: 1000
gidNumber: 1000
HomeDirectory: /home/kissir

Mindig az első sorban található az adatelem teljes DN-je. Utána jönnek az attribútum név és érték párok kettősponttal elválasztva. Ezeknek a sorrendje nem meghatározott. A többszörös értékű attribútumok többször vannak felsorolva. Az attribútum nevekben a nagy és kisbetűket nem különböztetjük meg egymástól.

LDAP vs. SQL

Az első bekezdésben megfogalmazott célokat egy SQL adatbázissal is meg lehet valósítani, így felmerülhet a kérdés, hogy miért jobb mégis LDAP-val csinálni? Az alábbiakban felvillantok néhányat a lehetséges okok közül, melyek nem jelentenek perdöntő érveket, de a rendszertervezésnél figyelembe veendőek:

- **LDAP keresésre optimalizált**: míg az SQL adatbázisok általános célú felhasználásra, addig az LDAP szerverek elsősorban ritkán változó adatok keresésére optimalizáltak. A felhasználói azonosítás, személyek és eszközök nyilvántartása stb. mind a saját életciklusukon belül ritkán változó adatokat szolgáltatnak. A legtöbb feladat pedig az ilyen adatbázisokon való kifinomult kereséssel oldható meg.
- **LDAP kifinomultabb azonosítási és hozzáférés szabályozási mechanizmussal rendelkezik**: maga az X.500-as szabvány is, de az LDAP protokoll is tartalmazza az un. bindig mechanizmust, melynek segítségével akár minden DIT-beli azonosíthatja magát, mielőtt bármilyen műveletet végezne az adatbázison. Ez a bindig mechanizmus ráadásul lehetővé teszi, hogy az azonosítást úgy végezzük el, hogy az adatbázisból az azonosítási token (pl.: jelszó) nem kerül ki soha. Mindehhez az LDAP szerverek többsége kifinomult hozzáférés-szabályozási mechanizmust is tartalmaz, aminek a segítségével akár minden egyes LDAP bejegyzésről megadhatjuk, hogy kik milyen műveletekhez férhetnek hozzá vagy akár azt is, hogy az adott attribútumnak milyen értéket adhatnak.

  Ezzel szemben általában a SQL alapú azonosítási mechanizmusok közönséges adatbázisban tárolják a jelszavakat (akár kódolva, akár kódolatlanul), az SQL kliens ezt a jelszót egy SQL lekérdezéssel kiolvashatja és összehasonlítja a felhasználó által megadottal. Maga az SQL kliensnek olyan adatbázis-felhasználóként kell a szerverhez kapcsolódnia, aki minden felhasználó adatához is hozzáfér.

- **LDAP alapvetően elosztott információs modellel rendelkezik**: míg egy SQL szerver esetén az elosztott működés meglehetősen bonyolultan valósítható meg, addig a DIT alapvetően könnyen eloszthető és ezt az elosztottságot a legegyszerűbb LDAP szerverek is támogatják. Ezen túlmenően az LDAP szerverek legalább master-slave jellegű, de nem ritkán muti-master, replikációs mechanizmussal rendelkeznek, amihez ráadásul nem szükséges közös elosztott állományrendszer sem.

Néhány esetben, például amikor már van egy SQL adatbázisunk felhasználói adatokkal, értelmes lehet a két módszert kombinálni, azaz LDAP protokollon keresztül elérhetővé tenni az SQL adatbázis adatait azonosítási célból.

Fejlődés

Számos címtárszolgáltatás létezett korábban is és létezik továbbra is, UNIX alatt ugyanúgy, mint és más rendszereken. A Novell NDS, a Microsoft ADS, a Banyan StreetTalk és az OSI-szabvány X.500, csak hogy néhány példát említsek. Az LDAP célja eredetileg a DAP – az X.500 címtárhozzáférési protokolljának – leegyszerűsítése volt.
Az LDAP a DAP radikálisan leegyszerűsített, TCP/IP alá készült változata. Az LDAP célja az volt, hogy megtartsa a DAP legfontosabb részeit, de elhagyjon mindent, ami felesleges: az X.500 bejegyzések hierarchiájának megőrzése mellett az LDAP használatával erőforrások takaríthatók meg. A TCP/IP használata pedig jelentősen leegyszerűsíti az alkalmazás és az LDAP-szolgáltatás közötti felületek kialakítását.

Időközben az LDAP továbbfejlődött és egyre több helyen alkalmazták önálló megoldásként, X.500 támogatás nélkül. Az LDAP 3-as verziója (LDAPv3, az openldap2 csomag által is támogatott protokollverzió) már támogatja a *hivatkozások* (referral) használatát, ami lehetővé teszi elosztott adatbázisok készítését. Az SASL (simple authentication and security layer, egyszerű hitelesítési és biztonsági réteg) használata szintén új.

Az LDAP már régóta nincs arra korlátozva, hogy az adatokat X.500 címtárszerverekről kérje le, mint azt eredetileg tervezték. Létezik egy nyílt forráskódú szerver, az slapd, ami képes az objektumadatok tárolására egy helyi adatbázisban. Van továbbá egy slurpd nevű kiegészítés is, amely több LDAP-szerver replikálásáért felelős. Mára már a korábban említett címtárak is kivétel nélkül biztosítanak LDAP-n keresztüli hozzáférést.

Az openldap2 csomag részei:

slapd

: Egy önálló LDAPv3-szerver, amely egy BerkeleyDB alapú adatbázisban tárolja az objektumadatokat.

slurpd

: Ez a program lehetővé teszi a helyi LDAP-szerveren lévő adatok módosításainak átvezetését (»replikálását«) a hálózat többi LDAP-szervereire.

LDAP vagy NIS?

UNIX-rendszereken hagyományosan a NIS nevű szolgáltatást használják a névfeloldáshoz és az adatok szétosztásához a hálózatban. Az /etc könyvtárban lévő fájlok és a group, hosts, mail, netgroup, networks, passwd, printcap, protocols, rpc, valamint services könyvtárak konfigurációs adatait a kliensek osztják szét a hálózatban. Ezek a fájlok könnyedén karbantarthatók, hiszen egyszerű szövegfájlok. Nagyobb mennyiségű adat kezelése azonban jóval bonyolultabb, ugyanis a NIS-ben nincs különösebb strukturáltság. A NIS-t kifejezetten UNIX-platformokhoz tervezték, ezért nem használható központi adatkezelőként egy heterogén hálózatban.

Szemben a NIS-sel, az LDAP használata nem korlátozódik tisztán UNIX-hálózatokra. A windowsos szerverek (a Windows 2000 óta) támogatják az LDAP felhasználását címtárszolgáltatásként. A Novell szintén biztosít LDAP-szolgáltatást. A fent említett alkalmazásfeladatok nem UNIX-rendszereken is támogatottak.
Az LDAP alapelve minden központilag felügyelendő adatstruktúrára alkalmazható. Néhány példaalkalmazás:

- Alkalmazás a NIS szolgáltatás kiváltására
- Levéltovábbítás (postfix, sendmail)
- Címjegyzék levelezőprogramokhoz (például Mozilla, Evolution vagy Outlook)
- BIND9 névszerver zónaleírásának adminisztrációja

A lista bővíthető, ugyanis az LDAP, szemben a NIS-sel, szabadon kiterjeszthető. A világosan definiált adatstruktúra leegyszerűsíti a nagy adathalmazok kezelését, mert könnyebbé és hatékonyabbá teszi a kereséseket.

PKI

Korunk informatikájának kulcs kérdésévé vált az adatbiztonság és a hitelesség. A számítógépes hálózatok fejlődése (Internet), az elektronikus kereskedelem és pénzforgalom kialakulása, olyan adatbiztos környezetet igényelnek, mely könnyen kezelhető és törvényileg elfogadott. Az egyszerű kezelhetőséget az elmúlt néhány év technológiai fejlődése valósította

meg, míg a törvényességi keretet az a világszerte beindult törvénykezési folyamat, ami a digitális aláírás révén hivatalossá teszi az elektronikus dokumentumokat.

A PKI rendszer, mint azt neve is mutatja (Public Key Infrastructure), olyan alkalmazás környezetet (infrastruktúrát) kínál, ami lehetővé teszi a törvények által elfogadott kétkulcsú harmadik személyes hitelesítési és adatbiztosítási eljárások használatát a számítógépes alkalmazások számára.

A PKI rendszerek alkalmazásának előnyei nem csak abból a kötelezettségből származnak, hogy használatát a jövőben akár törvényileg is elő lehet írni, noha önmagában nagy előny jelent az, ha valaki olyan rendszert alkalmaz, ahol az elektronikus dokumentumok (pl.: banki dokumentumok, megrendelések, számlák, államigazgatási dokumentumok, stb.) hivatalosan elismertethetők. A PKI rendszerek kialakításával, a chip kártyák használatával a számítástechnikai környezet és az egyre terjedő adatátviteli rendszerek (pl.: Internet) úgy tehetők biztonságossá és hitelessé, hogy kezelésük továbbra is egyszerű marad. Sok esetben ezen egyszerűség a mai biztonságos alkalmazáskezelést egyszerűsíti például akkor, ha több alkalmazás vagy elektronikai rendszer password-jei vagy kártyái helyett, egy chip kártyát lehet használni. Mindez jelentős költségcsökkenést jelent, ha a párhuzamos biztonsági rendszerek password vagy kártya adminisztrációja helyett egy PKI alapú chip kártya adminisztrációt kell üzemben tartani. Az adminisztráció csökkenése természetesen a biztonságot is növeli.

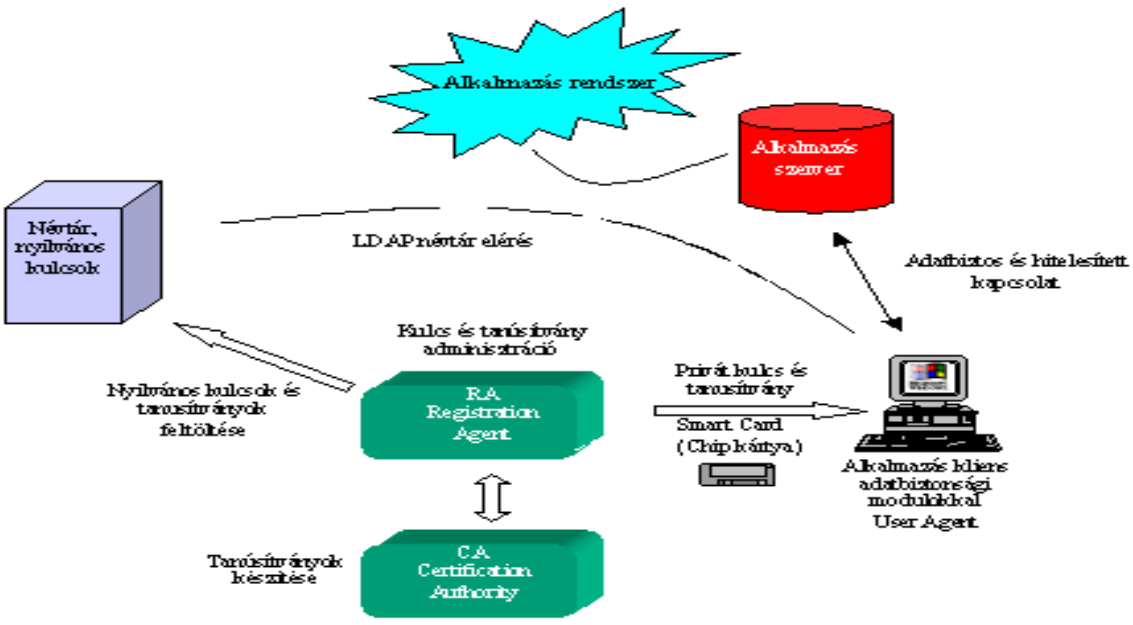
Az elektronikus adatok hitelesítése és kódolása egységes alapelven működik, amit a technológiában PKI rendszernek nevezünk. A rendszerben a felek két kulccsal (kóddal) rendelkeznek. Az egyik kulcs a saját, privát kulcs, amit csak az adott fél ismer és birtokol (pl.: smart kártyán). A másik kulcs a nyilvános, public, amit mindenki elérhet, és általában elektronikusan jól elérhető helyen tárolnak (pl.: névtárakban). A két kulcs kombinációjával digitális aláírást lehet képezni, illetve adattartalmat lehet kódolni, titkosítani úgy, hogy a másik félnek a hitelesség ellenőrzésére, illetve az adattartalom dekódolására csak a küldő fél nyilvános kulcsára van szüksége. A privát kulcs minden esetben továbbra is titkos marad, ugyanakkor a hitelesített és adatbiztos tranzakció elvégezhető. Lényeges követelmény, hogy a saját és nyilvános kulcsok hitelesek legyenek, vagyis hogy annak valódiságát és az adott félhez való tartozását hiteles harmadik fél garantálja, közjegyezze. E feladatot az elektronikus közjegyző végzi a rendszerben (CA= Certification Authority). A CA mint harmadik fél nem tesz mást,

mint aláírja, hitelesíti az adott fél kulcsait, amit tanúsítványként (certification) szolgáltat a kulcsokhoz.

A PKI alkalmazások jelentős mértékben függnek az őket támogató címtárszolgáltatástól. A címtár biztosítja a tanúsítványok tárolásához és szétküldéséhez szükséges eszközöket, gondozza és karbantartja a tanúsítványokat. A címtár szerverek általában az X.500 szabvány vagy alszabványainak megvalósításai.
Az X.500 egy sor ajánlásból áll, és maga a specifikáció is több ISO szabványra hivatkozik. Olyan címtárszolgáltatáshoz dolgozták ki, amely rendszer, szervezet vagy akár határon átnyúló használatra is alkalmas. Az X.500 egy protokoll készletet határoz meg az olyan műveletekhez, mint láncolás (chaining), nyomon követés (shadowing), a szerver-szerver kommunikációban történő irányítás (referral) és a Címtár Hozzáférési Protokoll (DAP, Directory Access Protokol) a kliensszerver kommunikációban.
A jelenleg az LDAP protokoll terjedt el a DAP alternatívájaként. A legtöbb címtár szerver és kliens támogatja a LDAP-t, míg a bonyolultabb DAP-ot nem mindegyik támogatja

Általános felépítés, működés

A PKI rendszer általános felépítését az alábbi ábra mutatja be:



A privát és a nyilvános kulcsok létrehozása több helyen is történhet. A létrehozó lehet a CA, RA is, illetve a kulcsok a chip kártyán is generálhatók, aminek az az előnye, hogy a privát kulcs sohasem hagyja el a smart (chip) kártyát. A nyilvános kulcsok a névtárban kerülnek tárolásra, a privát kulcsok smart kártyán vagy egyéb hordozón kerülnek a felhasználókhoz. A kulcspárok, felhasználók szerinti kezelését az RA–val lehet szakszerűen adminisztrálni, ami biztosítja a nyilvános azonosítók és tanúsítványok tömeges továbbítását is a névtárba. A CA tanúsítványokat állít elő, amivel harmadik személyként hitelesíti a nyilvános és privát kulcsok tulajdonoshoz való tartózását. A tanúsítványkészítési kérelmeket az RA indítja a CA felé, a CA az adott kulcsokhoz elkészített tanúsítványokat az RA felé továbbítja. A tanúsítványok a nyilvános és privát kulcsokhoz mellékelődnek.

Az alkalmazás kliens oldalon, a privát és nyilvános kulcsok elérését, összevetését és a rejtjelezést biztonsági modulok végzik. Az alkalmazásfunkciók kezelése mellett lehetőség nyílik az elektronikus dokumentumok digitális aláírására és titkosítására is. Az alkalmazás kliensek, a nyilvános kulcsok kezelésekor rendszerint LDAP protokollon keresztül érik el a névtárat.

A hitelesítésben résztvevő valamennyi felhasználó tanúsítványozott nyilvános kulcsa tárolásra kerül a névtárban.

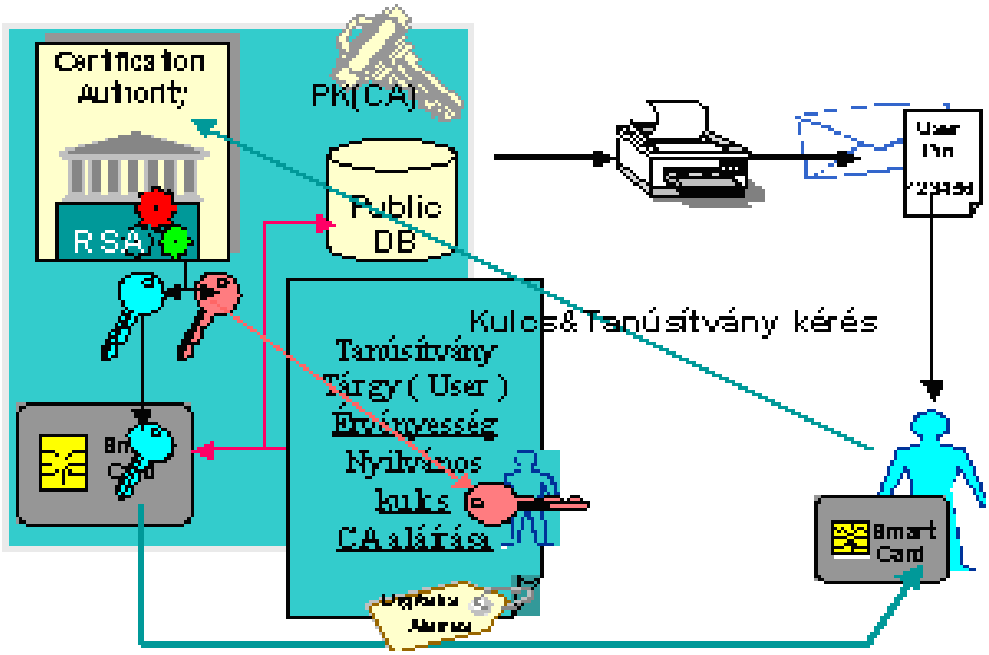
Hitelesítéskor az alkalmazás kliens chip kártyáról olvassa be a privát kulcsot, LDAP protokollon keresztül éri el a névtárban lévő nyilvános kulcsot, és ezekkel a kulcsokkal különféle kombinációk szerint digitális aláírást képez. A kulcsok kombinálásával, a tartalmat is rejtjelezni (titkosítani, kódolni) lehet.

Fogadó oldalon, az alkalmazás kliens vagy szerver a saját nyilvános és privát kulcsa, valamint a küldő nyilvános kulcsának kombinálásával a rejtjelezett tartalomhoz hozzáfér, illetve a digitális aláírás hitelességét ellenőrzi. Az alkalmazás bármilyen jellegű lehet, a hitelesítés és titkosított adatkezelés az alkalmazás használói között történik. WEB-es alkalmazáskor a hitelesítési és titkosítási eljárások hasonló elvek szerint működnek, csak a szereplők közötti adatbiztos és hitelesített kapcsolatot, az on-line technológiákra jellemző adatbiztonsági és hitelesíti modulok biztosítják.

Kulcsok generálása

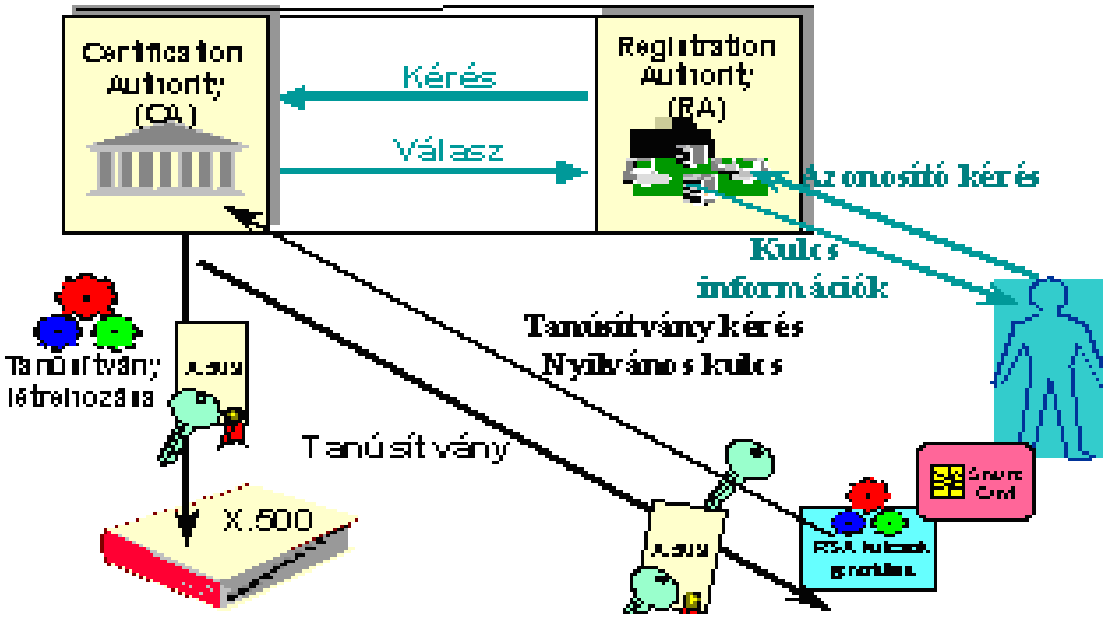
Alapvető biztonságtechnikai kérdés a kulcsok előállítási helye, ami valamelyik PKI elemen történik. A PKI rendszerek fejlődése során ez egyre több elemen vált lehetővé. Általánosságban két lehetőség kínálkozik a kulcspárok generálására, és mindkettő rendelkezik mind a maga előnyeivel, mind hátrányaival a CA illetve a felhasználó oldaláról nézve. A két lehetőséget centralizált és decentralizált kulcsgenerálásnak nevezzük.

Centralizált kulcspár generálás folyamata



Lényegét tekintve, központi (centralizált) kulcsgenerálásról beszélünk, amennyiben a kulcsokat a CA oldalon, illetve lokális kulcs előállításról (decentralizált), amennyiben a felhasználó vagy a RA oldalán állítjuk elő.

Lokális kulcspár generálás folyamata



A kulcsok generálásának jelenlegi legbiztonságosabb módja, ha olyan chip kártyán tároljuk a kulcsokat, amely egyben azok generálására is képes. Ezt a módszert nevezi az irodalom on-board kulcsgenerálásnak. A módszer biztonsága abból adódik, hogy az előállított kulcsok közül az egyén titkos kulcsa soha nem tartózkodik a kártya memóriáján kívül. A nyilvános kulcs a többi módszerhez hasonlóan kerül a CA-hoz. Minden valószínűség szerint biztonságossága miatt, ez utóbbi módszer fog a közeljövőben elterjedni.

PKI rendszer eleme

A gyakorlatban, a PKI elemei sokszor fontos alkalmazásintegrációs elemeket jelentenek, így a PKI bevezetése szorosan érintheti egy szervezet meglévő és a jövőben kialakulásra kerülő elektronikus dokumentumkezelő alkalmazásait.

PKI elemként soroljuk fel a névtárakat is. Az utóbbi idők LDAP integrációs és névtár technológiai fejlődése a névtárakat egyre inkább előtérbe helyezte a nyilvános kulcsok és tanúsítványok tárolására és elérésére. A névtár technológia önálló technológiának tekinthető és az utóbbi időben szintén jelentős alkalmazásintegrációs szerepre tett szert. A PKI rendszerek elterjedt alkalmazás környezetének tekinthető.

A rendszer telepítése

Én az LDAP – Server futtatásához egy az internetről letölthető Linux disztribúciót, az OpenSUSE 10.2 –t használtam, mely tartalmazza az LDAP – Servert is. Az operációs rendszer több CD-s vagy egy DVD-s változata letölthető az alábbi webcímről:

http://hu.opensuse.org/Kiadott_változat#Let.C3.B6lt.C3.A9s .

Az OpenSUSE telepítését itt nem részletezem, csak az LDAP-t tartalmazó csomagok bemutatását tekintem célnak.

Lépésről lépésre

1. lépés - Az LDAP telepítése YAST2-vel

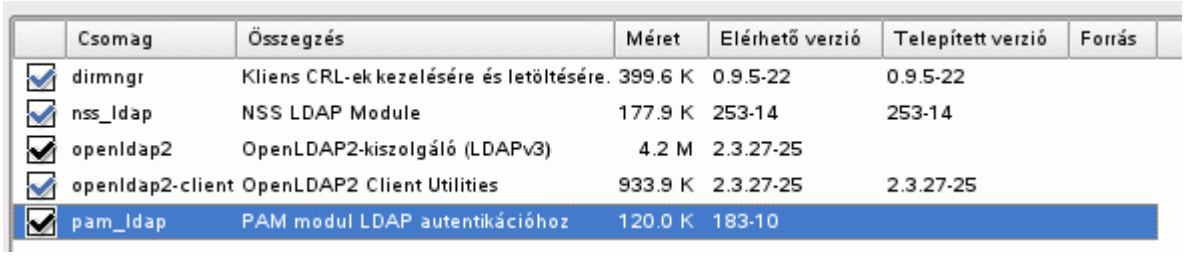
Az LDAP-t támogató csomagok telepítéséhez a szintén az operációs rendszer részét képező YAST2-t használtam, mely grafikus felületen teszi lehetővé csomagok telepítését és törlését. Ehhez válasszuk a

Szoftver -> Szoftver telepítése, eltávolítása

menü pontokat. Majd a

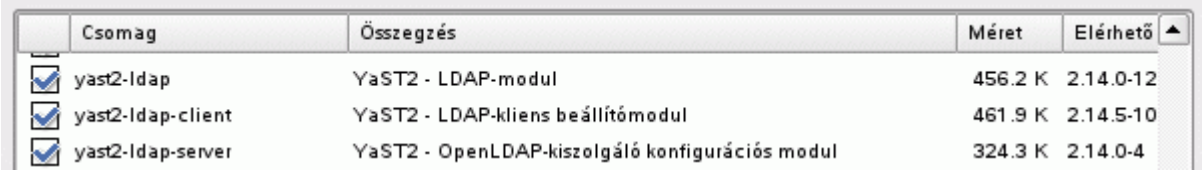
Hálózat -> LDAP

csomag csoportból jelöljük meg az alábbi csomagokat telepítésre:



Csomag	Összegzés	Méret	Elérhető verzió	Telepített verzió	Forrás
dirmngr	Kliens CRL-ek kezelésére és letöltésére.	399.6 K	0.9.5-22	0.9.5-22	
nss_ldap	NSS LDAP Module	177.9 K	253-14	253-14	
openldap2	OpenLDAP2-kiszolgáló (LDAPv3)	4.2 M	2.3.27-25		
openldap2-client	OpenLDAP2 Client Utilities	933.9 K	2.3.27-25	2.3.27-25	
pam_ldap	PAM modul LDAP autentikációhoz	120.0 K	183-10		

Szükségünk lesz a YAST2 LDAP konfigurálását támogató alábbi csomagok által tartalmazott modulokra is.



Csomag	Összegzés	Méret	Elérhető
yast2-ldap	YaST2 - LDAP-modul	456.2 K	2.14.0-12
yast2-ldap-client	YaST2 - LDAP-kliens beállítómodul	461.9 K	2.14.5-10
yast2-ldap-server	YaST2 - OpenLDAP-kiszolgáló konfigurációs modul	324.3 K	2.14.0-4

2. lépés - LDAP adatbázis létrehozása

Az LDAP – Server konfigurálásához válasszuk a

Hálózati szolgáltatások -> LDAP – kiszolgáló

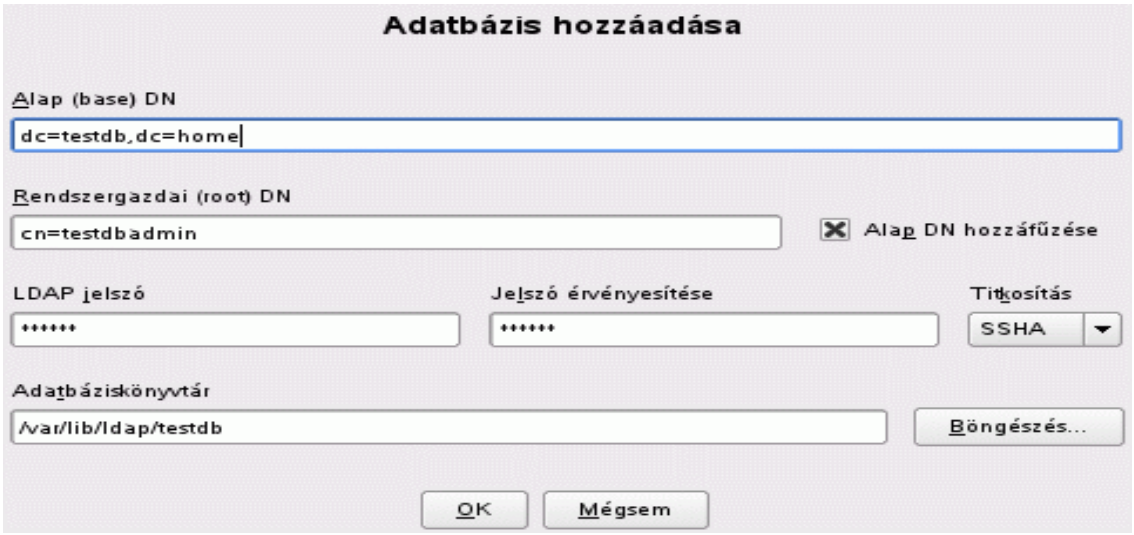
menü pontot. Itt elindítható az LDAP – Server és annak beállításai is elvégezhetőek.

LDAP – kiszolgáló beállítások

Az LDAP - kiszolgáló indítása (igen vagy nem), pontban elindíthatjuk, vagy leállíthatjuk az LDAP - kiszolgálót. Ha azt szeretnénk, hogy az LDAP szervert külső hálózatról is el tudjuk érni, akkor a „Tűzfal portjának megnyitása” opciót be kell kapcsolni. A „Tűzfalbeállítások” gombra klikkelve további lehetőségeket adhatunk, vagy tilthatunk le a külső felhasználók előtt.

A „Beállítások” gombra kattintva az LDAP - kiszolgáló állítható be.

Adatbázis hozzáadása



Többek között beállítandó az LDAP által használt adatbázis is.

Új adatbázis hozzáadásához a fenti adatok kitöltése szükséges vagy a /etc/openldap/slapd.conf állomány alábbi módosítása (részlet az fájlból):

```
#######################################################################
#########
# BDB database definitions
#######################################################################
#########

loglevel 0
database bdb
suffix "dc=testdb,dc=home"
rootdn "cn=testdbadmin,dc=testdb,dc=home"
rootpw "{ssha}FXOaMTIicUN3CMe8W0+Rokl2O2JVUlVGSg=="
directory /var/lib/ldap/testdb
checkpoint 1024 5
cachesize 10000
index objectClass,uidNumber,gidNumber eq
index member,mail eq,pres
index cn,displayname,uid,sn,givenname sub,eq,pres
```

3. lépés - Az LDAP - kliens beállítása

Hálózati szolgáltatások -> LDAP - kliens

Az LDAP kliens beállítása

A felhasználók LDAP - kiszolgálón keresztül való azonosításához, válasszuk ki az „LDAP használata” pontot. Az NSS és a PAM ennek megfelelően lesz beállítva.
Az LDAP szolgáltatás kikapcsolása az „LDAP leállítása” pont kiválasztásával tehető meg. Ha letiltjuk az LDAP-t, akkor a YaST törli az /etc/nsswitch.conf fájl jelenlegi LDAP bejegyzéseit. A PAM beállítása is módosul, az LDAP - bejegyzések kikerülnek onnan.

Abban az esetben, ha szeretnénk az LDAP-t bekapcsolni, de a felhasználókat kitiltani a gépről, válasszuk ki az „LDAP használata letiltott bejelentkezésekkel” lehetőséget.

Ezután adjuk meg az LDAP - kiszolgáló címét (pl. localhost vagy 11.2.0.2) az első bejegyzésben, a keresési alap ("base DN", pl. cn=testdbadmin, dc=testdb, dc=home) megkülönböztetett nevét a másodikban.

Több kiszolgáló megadása esetén azok adatait szóközökkel kell elválasztani. A címek feloldásának LDAP nélkül is mennie kell. Megadhatjuk továbbá a portot, amelyen a kiszolgáló fut, a következő szintaxissal: "kiszolgáló:port", például localhost:23.

A „Keresés” gombbal válasszuk ki az LDAP - kiszolgálót az SLP-n keresztül meghirdetett listából.

A „DN lekérése” paranccsal olvasható ki az alap DN a kiszolgálóról.

Néhány LDAP - kiszolgáló támogatja a StartTLS [RFC2830] szabványt. Ha a milyenk ilyen és a protokoll aktív, jelöljük be az „LDAP TLS/SSL” lehetőséget az LDAP adatok titkosított továbbításához. Alapesetben az LDAP protokoll 3-as verziója kerül beállításra. Ha 2-es protokollt használó LDAP - kiszolgálónk van (például OpenLDAP v1), állítsuk be a „2-es LDAP verzió” használatát.

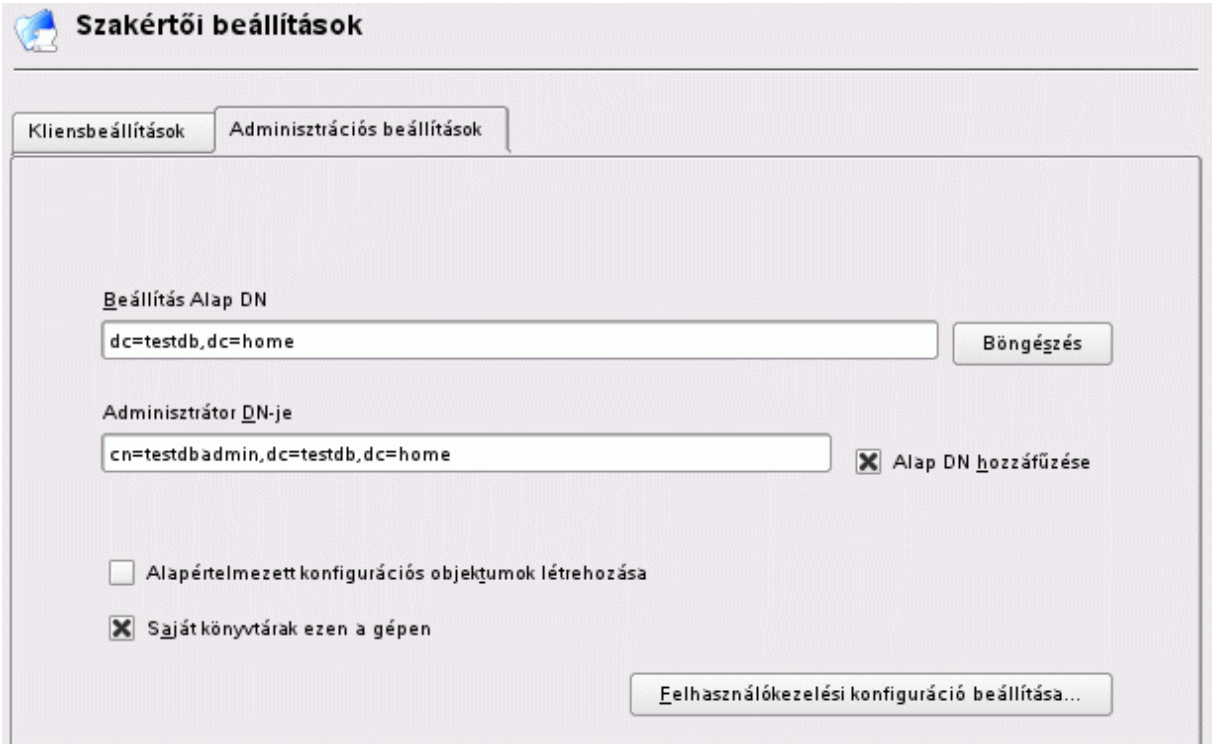
Az LDAP - kiszolgáló szakértői beállításaihoz kattintsunk a „Szakértői beállítások gombra”.

Könyvtárak automatikus csatolásához válasszuk „Az automounter aktiválását”. Ekkor feltételezzük, hogy ennek konfigurációs fájljai (auto.*) már léteznek, vagy helyileg, vagy LDAP-n keresztül. Ha nincs telepítve, de szeretnénk használni, akkor a csomag automatikusan telepítésre kerül.

Szakértői beállítások / Kliensbeállítások

Az egyes leképezésekhez adjuk meg (felhasználók, jelszavak és csoportok) a használt keresési alapokat, ha azok nem egyeznek meg az alap DN-nel. Ezek az értékek az /etc/ldap.conf fájl nss_base_passwd, nss_base_shadow és nss_base_group attribútumaiba kerülnek elmentésre.
A „Jelszóváltási protokoll” az /etc/ldap.conf fájl pam_password attribútumára utal. Válasszunk, hogy milyen típusú LDAP - csoportokat használunk. A „Csoporttag attribútum” alapértelmezett értéke: member.

Szakértői beállítások / Adminisztrációs beállítások



Itt állíthatjuk be a hozzáférést a kiszolgálóhoz.

A „Beállítás Alap DN” az alap DN (Distinguished Name) az LDAP - kiszolgálón tárolt adatok alapja.

A kiszolgálón tárolt adat eléréséhez adjuk meg az „Adminisztrátor DN-jét”. Megadhatjuk a teljes DN-t (például: cn=testdbadmin, dc=testdb, dc=home) vagy csak a relatív DN-t (például: cn=testdbadmin). Az „Alap DN hozzáfűzésének” engedélyezésével az LDAP alap DN hozzá lesz fűzve automatikusan.

Az „Alapértelmezett konfigurációs objektumok létrehozása” teszi lehetővé az LDAP felhasználók és csoportok alapértelmezett konfigurációs objektumainak létrehozását. Az objektumok csak akkor kerülnek létrehozásra, ha még nem léteznek.

A „Felhasználókezelési konfiguráció beállítása…” gombbal az LDAP - kiszolgálón tárolt beállítások módosíthatók. Ha nincs még kapcsolat az LDAP - kiszolgálóval, vagy módosítottuk a beállításokat, akkor meg kell adni az adminisztrátori jelszót.

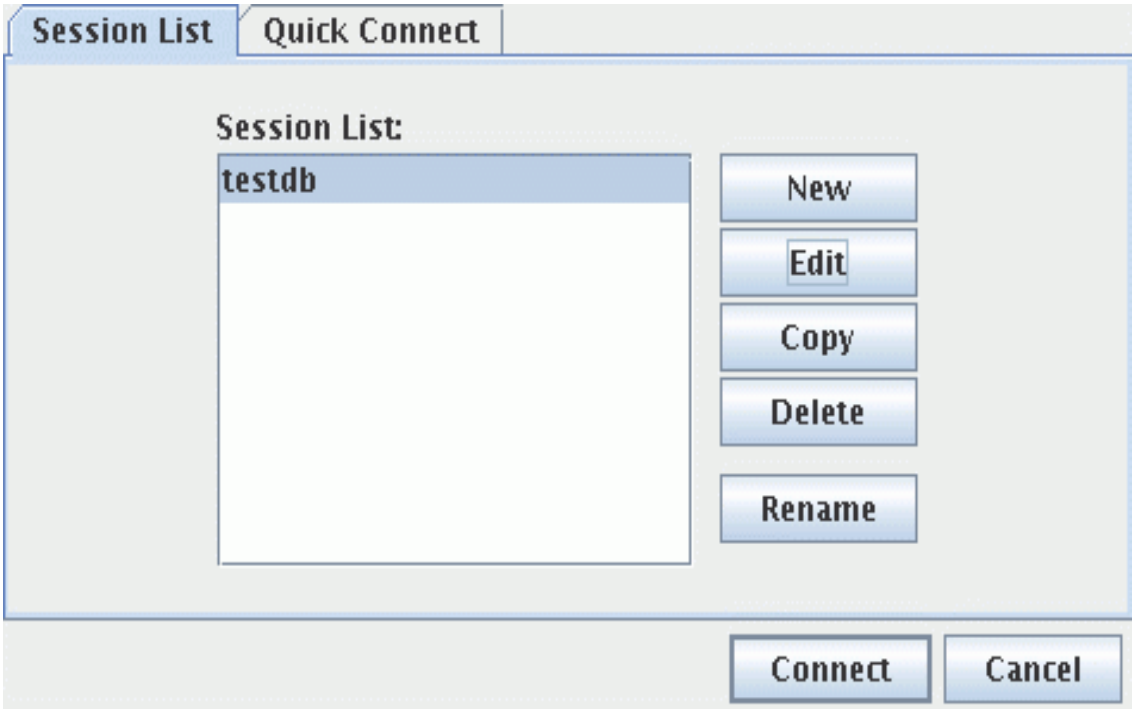
A „Saját könyvtárak ezen a gépen” bejelölésével a felhasználók saját könyvtárai ezen a gépen tárolódnak. Az érték módosításának nincs közvetlen hatása, azonban a YaST felhasználói modul számára fontos, amellyel kezelhetők a felhasználók saját könyvtárai.

4. lépés – Letöltött – browser

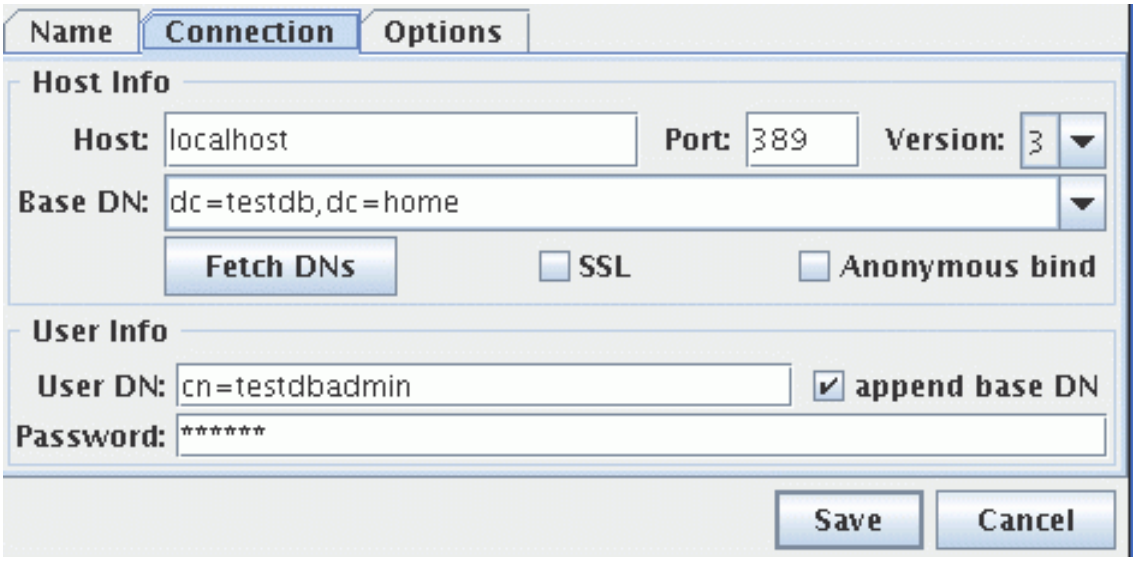
Az LDAP szerverhez több letölthető böngésző is található a weben. Egy részük komplett user szolgáltatást nyújt, mint például az LDAP Account Manager. Ezek beállítása, és konfigurálása nem része a diplomamunkámnak, de igen hasznos lehet LDAP szerveren keresztül történő autentikációhoz. Ehhez azonban szükséges a Samba megfelelő konfigurálása.
Az LDAP-on keresztül történő autentikációhoz szükséges beállításokat nem részletezem, hanem az LDAP felépítésére és használatára reflektálok, így a továbbiakban inkább egy Windows XP rendszeren is használható LDAP – böngészőt mutatok be.

Az LDAP Browser egy java-ban írt verziója letölthető a következő web címről: **http://www-unix.mcs.anl.gov/~gawor/ldap/download.html**. Ez a böngésző használható Linux és Windows platformokon is. A letöltött tömörített állomány kicsomagolása után futtatható, ha az adott gépen van java.

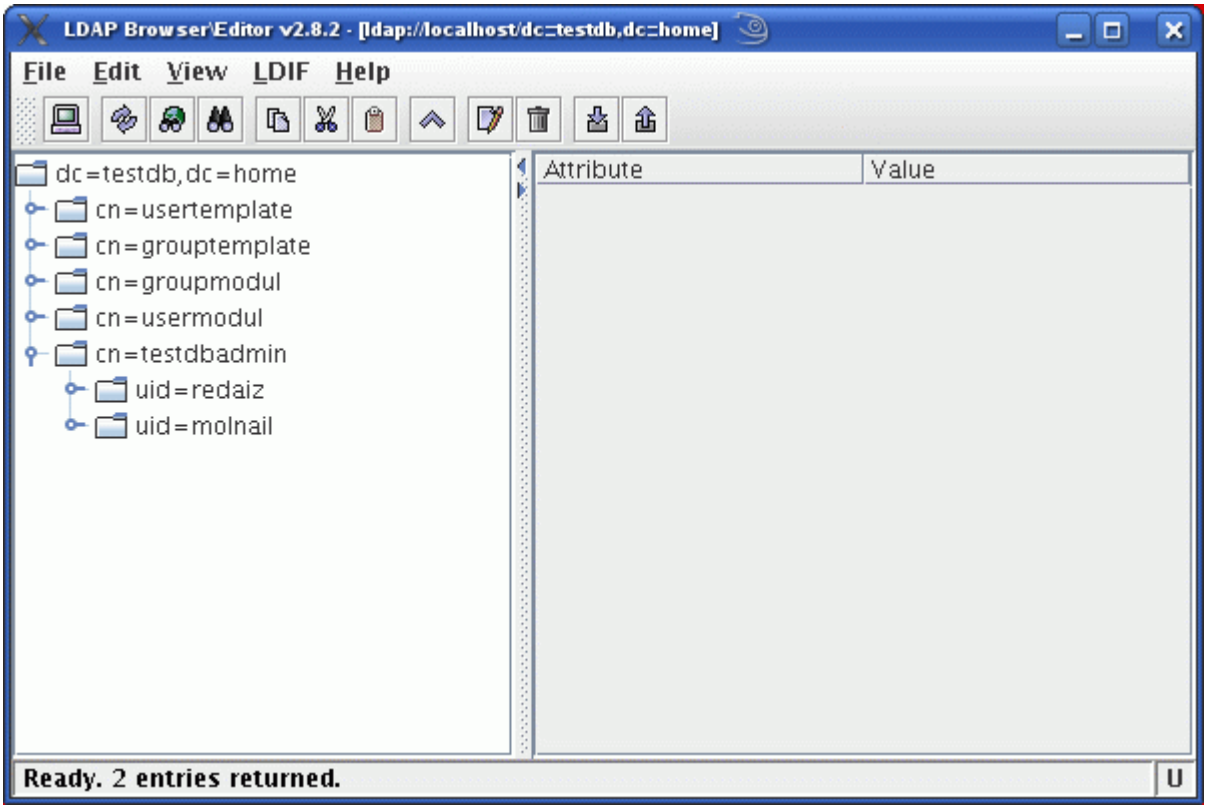
A browser indítása: lbe.sh



Az indítás után megjelenő ablakban az LDAP – Server adatainak megadása szükséges a kapcsolathoz.



Kapcsolódás után az alábbi felület számos lehetőséget biztosít az LDAP-ban tárolt adatok kezeléséhez.

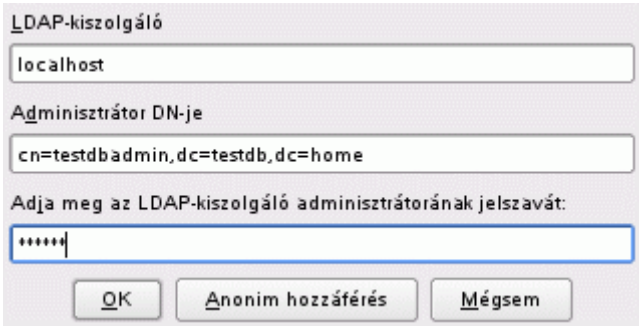


5. lépés - A YAST LDAP - Browser beállítása

Ha a fent említett LDAP – Browser letöltésére nincs lehetőségünk vagy egyszerűen csak nem akarunk vele fáradozni, akkor használhatjuk az OpenSUSE részét képező LDAP – böngészőt is, mely elérhető a YAST2-ben a

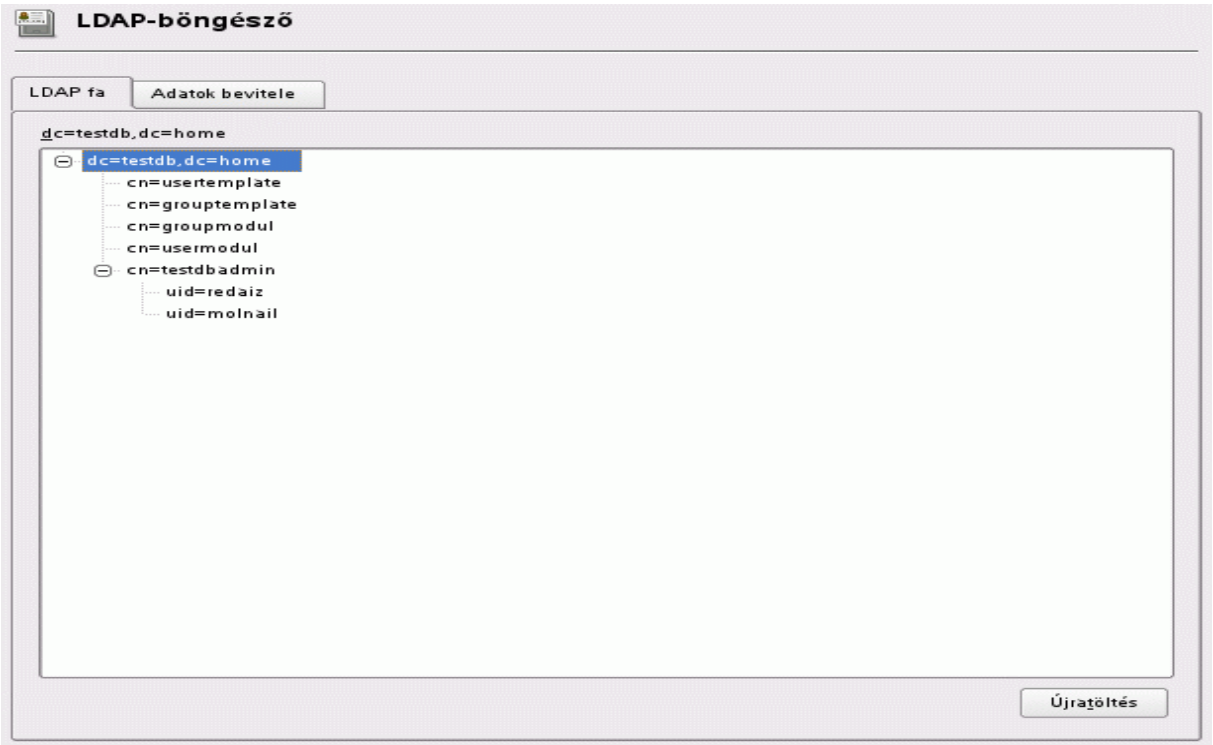
Hálózati szolgáltatások -> LDAP böngésző menü pont alatt

LDAP - böngésző / Adatok bevitele



Ezen böngésző számára is szükséges az LDAP – Server adatainak megadása.

LDAP - böngésző / LDAP – fa



Az LDAP címtárfa az LDAP fa lapon járható be. A kiválasztott LDAP objektum adatai módosíthatóak az „Adatok bevitele” lapon.

6. lépés – Új LDAP – felhasználó megadása

Az LDAP adatbázisába új, felhasználó adatait tartalmazó objektum felvehető a YAST2

Biztonság és felhasználók -> Felhasználók

menü pont alatt is. Ehhez szükséges megadnunk, hogy LDAP felhasználót szeretnénk létrehozni.

Adjuk meg a felhasználóra vonatkozó Keresztnév, Családi név, Azonosító és Jelszó mezőket.

A jelszó megadásakor a kis- és nagybetűket meg kell különböztetni, és a jelszó lehetőleg ne tartalmazzon semmilyen speciális karaktert (pl. ékezetes betűt).

A kiválasztott titkosítási eljárásnál (CRYPT) a jelszó legfeljebb 8 karakter hosszú lehet és legalább 5 karakterből kell állnia.

A jelszóban csak az angol billentyűzeten megtalálható karakterek használhatók. Rendszerhiba esetén előfordulhat, hogy nemzeti billentyűzetkiosztás nélkül kell bejelentkezni.

A jelszót kétszer kell megadni, hogy a félreírást kiküszöböljük. Ne felejtsük el jelszavunkat!

A felhasználónév előállítható a teljes névből a „Javaslat” gomb megnyomásával. Ezt a javaslatot módosíthatjuk, de csak (ékezet nélküli) betűkkel, számokkal és az ._- jellel. Nagybetűket csak akkor használjunk, ha pontosan tudjuk, mit csinálunk! Amint láthatjuk, a felhasználónevekre szigorúbb az előírás, mint jelszavakra. A korlátozások az /etc/login.defs fájlban írhatók át.

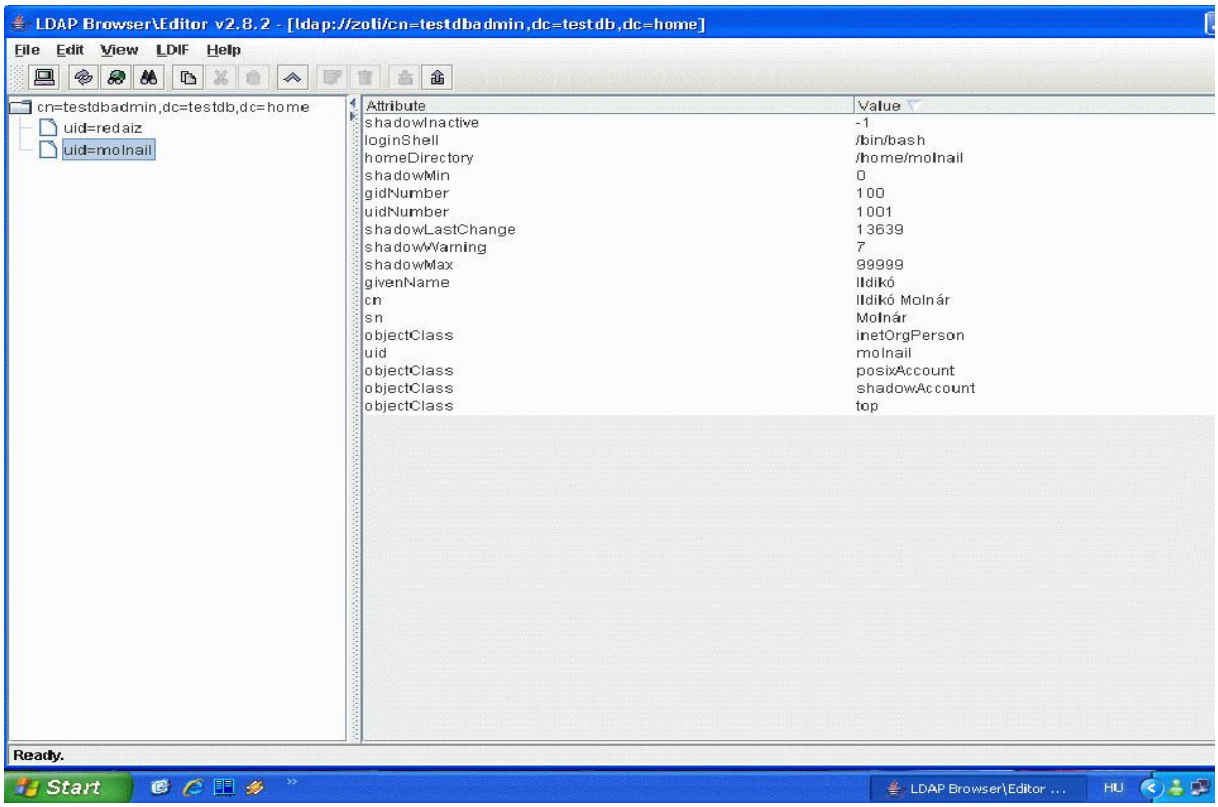
Ha a felhasználó beállításainak további részleteit (mint például a saját könyvtár vagy a felhasználói azonosító) is szeretnénk látni, nyomjuk meg a „Részletek” gombot.

A jelszóbeállítások (pl. a lejárati idő) megváltoztatásához kattintsunk a „Jelszóbeállítások” gombra.

Ha meg akarjuk tiltani ennek a felhasználónak, hogy bejelentkezzen, akkor jelöljük meg a „Felhasználó bejelentkezésének tiltása” négyzetet.

Az LDAP - szerverhez csatlakozás Windows XP-alól

A letöltött LDAP - Browser java alkalmazás, így Windows XP operációs rendszeren is futtatható. Az LDAP szerver elérésének megadása után semmiben nem különbözik a Linux alatt futtatott verziótól.



LDIF bejegyzés

Számos esetben hasznos lehet az LDAP adatok szöveges formátumú platform független reprezentációjára. Például különféle LDAP szerverek közötti adatmigráció az egyik tipikus felhasználás, de igen hasznos lehet, ha az LDAP szerver adatait emberi fogyasztásra szánjuk. Erre találták ki a LDAP Data Interchange Formatot (LDIF).

Egy konkrét példa egy minta LDIF bejegyzésre az alábbi:

molnail.ldif

```
dn: uid=molnail, cn=testdbadmin, dc=testdb,dc=home
shadowMin: 0
givenName:: SWxkaWvDsw==
sn:: TW9sbsOhcg==
userPassword:: e2NyeXB0fUdSYmxZYmlsSVF4aG8=
loginShell: /bin/bash
uidNumber: 1001
gidNumber: 100
shadowMax: 99999
uid: molnail
objectClass: top
objectClass: posixAccount
objectClass: shadowAccount
objectClass: inetOrgPerson
shadowLastChange: 13639
cn:: SWxkaWvDsyBNb2xuw6Fy
homeDirectory: /home/molnail
shadowInactive: -1
shadowWarning: 7
```

Irodalomjegyzék

Szakirodalmak:

Bill von Hagen, Brian K. Jones - Linux bevetés közben – Második küldetés

Peter Moulding: PHP haladóknak - Fekete könyv

Web címek:

https://nws.niif.hu/ncd2002/docs/ehu/50/index.html

http://www.gentoo.org/doc/hu/ldap-howto.xml

http://www.stanford.edu/~hodges/talks/mactivity.ldap.97/index2.html

http://www.umich.edu/~dirsvcs/ldap/doc/guides/slapd/1.html

http://padre.web.elte.hu/ldap.html

http://hu.opensuse.org