Hitelesítés LDAP használatával (3. rész)

Az LDAP-ról szóló sorozat záró írásában LDAP-kiszolgálónk Mick segítségével végre eljut odáig, hogy valódi felhasználókat valóságos környezetben is képes hitelesíteni.

Az elmúlt hónapokban egy LDAP-kiszolgálót építgettünk. Telepítettük az OpenLDAP-ot; beállítottuk a `slapd`-t, a kiszolgálódémont; életet leheltünk a TLS alapú titkosításba; végül létrehoztuk első LDAP-bejegyzésünket, egy fő szervezet bejegyzését. Elérkezett az idő, hogy a rendszerhez hozzáadjuk a felhasználókat, és IMAP-munkamenetek hitelesítésére használjuk a kiszolgálót.

Az adatbázis szerkezete

Az LDAP felhasználói adatbázisának létrehozásakor az első lépés a címtár szerkezetének kigondolása, amely annak eldöntését is magába foglalja, hogy a felhasználókat és az egyéb egyedeket kívánjuk-e csoportosítani, vagy egyszintű szerkezetet akarunk használni. Ha az LDAP-adatbázist kizárólag – és csakis kizárólag – hálózati telefonkönyv vagy hitelesítő kiszolgáló üzemeltetésére használjuk, akkor egy egyszintű adatbázis is megfelel. Ebben az esetben a felhasználók megkülönböztető nevei (Distinguished Name, DN) valahogy így alakulnak:

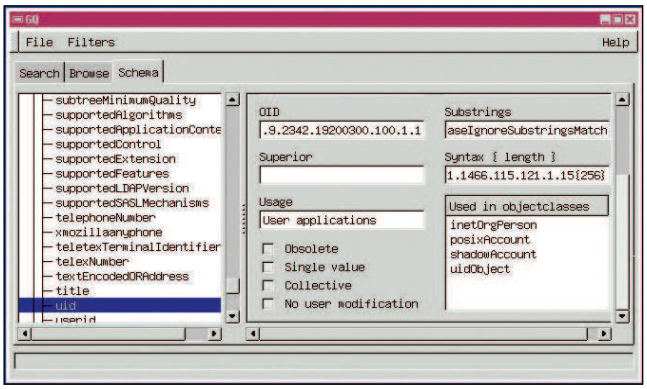
```
dn=Mick Bauer,dc=proba,dc=org
```

Ha azonban az adatbázis nemcsak személyekkel, de szervezeti csoportokkal, részlegekkel, számítógépekkel stb. kapcsolatos adatokat is tartalmaz, akkor kifinomultabb szerkezetű címtárra lehet szükség. Ilyet több módszerrel is lehet építeni. Az egyik lehetőség az, hogy a `domainComponent` (`dc`) mezőkkel tartományunkon belül altartományokat hozunk létre, függetlenül attól, hogy ezek a DNS-ben valóban léteznek-e. Ekkor a bejegyzések `dn=Bick Mauer,dc=tervezes,dc=proba,dc=org` formátumúak lesznek. Egy másik lehetőség az `organizationalUnit` objektumok hasonló módon való használata, ilyenkor `dn=Dick Lauer,ou=tervezes,dc=proba,dc=org` lesz a kapott formátum.

Annak érdekében, hogy a további mondandóm is követhető legyen, maradok az egyszintű adatbázis használatánál; természetesen senkinek nem szeretném a kedvét szegni: mindenki maga határozza meg, hogy igényeinek milyen szerkezetű LDAP-adatbázis felel meg a leginkább. A ➲ http://www.openldap.org címen elérhető és az OpenLDAP programhoz mellékelt leírásokban bőven találni példákat.

A séma és a felhasználórekordok

Szintén fontos döntés a használni kívánt és a rekordokba beillesztendő LDAP-jellemzők körének a meghatározása. A múlt alkalommal már szóltam arról, hogy ezek hogyan vannak csoportosítva a sémákban, illetve milyen kapcsolatban állnak egymással. Bizonyára mindenki emlékszik, hogy a */etc/openldap/slapd.conf* fájlban megadott sémák határozzák meg, hogy a rekordokban milyen jellemzőket lehet tárolni. A sémákat nem elég a */etc/openldap/slapd.conf* fájlban felsorolni: minden létrehozott rekordban a megfelelő sémákat `objectClass` állításokkal hozzá kell rendelni az egyes



A séma böngészése a gq segítségével

felhasználókhoz. Ugyancsak a múlt alkalommal mondtam el, hogy a */etc/openldap/schema* könyvtárban lévő sémafájlok határozzák meg, hogy milyen séma milyen jellemzőket tartalmaz, illetve hogy az adott sémán belül milyen objektumosztályokhoz milyen jellemzők tartoznak.

Tegyük fel, hogy az LDAP-kiszolgáló IMAP-kapcsolatokhoz fog hitelesítést végezni. Ekkor az `uid` és a `userPassword` LDAP-jellemzőkre mindenképpen szükség lesz. Hasonlók mondhatók el bármely más alkalmazásról is, ha az a Bind-eljárással végez LDAP alapú hitelesítést, vagyis hitelesítő programrésze egyszerűen megpróbál kapcsolódni az LDAP-kiszolgálóhoz úgy, hogy közben a felhasználó által megadott nevet és jelszót alkalmazza. Ha a csatlakozás sikeres, a program a hitelesítést sikeresnek ítéli, és lezárja az LDAP-kapcsolatot.

Azt, hogy az `uid` és a `userPassword` jellemzők mely sémában és objektumosztályokban találhatók meg, a legegyszerűbben úgy mérhetjük fel, hogy a `grep` paranccsal a */etc/openldap/schema* könyvtárban rákeresünk az `uid` és a `userPassword` karakterláncra, feljegyezzük a találatokat, kézzel végignézzük a megfelelő állományokat, és kikeressük az említett két jellemzőt `MUST()` vagy `MAY()` állításban tartalmazó objektumosztályokat. Jómagam az `uid` karakterláncra az OpenLDAP 2.0-t futtató Red Hat 7.3 rendszeren a következő fájlokban találtam hivatkozást: ***core.schema, cosine.schema, inetorgperson.schema, nis.schema*** és ***openldap.schema***.

A talált fájlokon a `less` segítségével végigszaladva a következőkre jutottam: a ***core.schema*** fájl `uidObject` objektuma megköveteli az `uid` jellemzőt, a ***cosine.schema*** egyetlen, az `uid` jellemzőre vonatkozó hivatkozása megjegyzésbe van téve, így érdektelen; az ***inetorgperson.schema*** egy `inetOrgPerson` objektumosztályt tartalmaz, amely kiegészítő jellemzőként támogatja az `uid` használatát, a ***nis.schema*** két objektumosztállyal rendelkezik – `posixAccount` és `shadowAccount` –, mindkettő megköveteli az `uid` elérhetőségét, nem különben az ***openldap.schema*** `OpenLDAPperson` objektumosztálya.

Szerencsére ezeket az adatokat sokkal gyorsabban is beszerezhetjük. A `gq` LDAP-eszköz segítségével LDAP-kiszolgálónk

1. lista LDIF fájl egy felhasználói rekord hozzáadásához

```
dn: cn=Wong Fei Hung,dc=proba,dc=org
cn: Wong Fei Hung
sn: Wong
givenname: Fei Hung
objectclass: person
objectclass: top
objectclass: inetOrgPerson
mail: wongfh@proba.org
telephonenumber: 651-344-1043
o: Proba
uid: wongfh
```

összes sémájának összes jellemzőjét végigböngészhetjük. *Képünkön* látható, hogy a saját LDAP-kiszolgálómon a `gq` szerint hol találhatók hivatkozások az `uid` jellemzőre. Képünkön látható Used in objectclasses (Ezekben az objektumosztályokban használva) lista elárulja nekünk, hogy a kiválasztott jellemző – jelen esetben az `uid` – az `uidObject`, a `posixAccount`, a `shadowAccount` és az `inetOrgPerson` objektumosztályokban tűnik fel, pontosan azokban, amelyeket a `grep` használatával is megtaláltunk. Az `OpenLDAPperson` objektumosztály nem tűnik fel a `gq` ablakában, mivel a kérdéses LDAP-kiszolgálón a */etc/openldap/slapd.conf* fájlban nincs olyan parancs, amely az ***openldap.schema*** fájl használatát előírná. Miután eldöntöttük, hogy milyen formátumú LDAP-rekordokat szeretnénk használni, bármikor eltávolíthatjuk a számunkra szükségtelen objektumosztályokat tartalmazó sémákat. Gondolom, mindez eléggé zűrös dolognak tűnik, és valóban az tud lenni, de túl kell esni rajta, hiszen olyan rekordokat kell létrehoznunk, amelyek képesek a környezetünk által igényelt adatok tárolására. Mivel az LDAP meglehetősen rugalmas, a jellemzők igényeinknek megfelelő összeválogatásához bizony szükség lehet egy kis bütykölésre.

Rekordok létrehozása és hozzáadása

Ahogy a sémák böngészése, úgy az LDAP-rekordok hozzáadása is végezhető kézzel és grafikus felületen egyaránt.
A múlt alkalommal kézzel hoztuk létre a fő szervezet bejegyzését, így első felhasználói rekordunkat is így fogjuk megalkotni. Az eljárás két lépésből áll: először készítenünk kell egy különleges, LDIF formátumú szövegfájlt, majd az `ldapadd` paranccsal be kell emelnünk a tartalmát az LDAP-adatbázisba. Vessünk egy pillantást az *1. listában* látható LDIF-fájlra.
Mivel minden mást ezek határoznak meg, az 1. lista elemzését a benne található `objectclass` állításokkal fogjuk kezdeni. Ez a felhasználó a `top` (minden rekord esetében kötelező), a `person` és az `inetorgperson` objektumosztályokkal van összerendelve. A `person` osztályt azért választottam ki, mert benne található meg a `userPassword` (ezt az 1. kódrészletben nem adtuk meg, ám Mr. Wong hamarosan jelszót is kap) és a `telephonenumber` jellemző – az utóbbira ugyan egyelőre nincs szükségünk, de a későbbiekben még lehet. Az `inetOrgPerson` objektumosztályban található az `uid` jellemző, valamint jó pár egyéb jellemző is, amelyek később még jól jöhetnek.
A sémában található `MUST` és `MAY` megszorításokat – többek között – úgy kerülhetjük meg, hogy a `schemacheck off` utasítással bővítjük a */etc/openldap/slapd.conf* fájlt. Így a ***slapd.conf*** fájlban hivatkozott sémafájlok bármelyikének bármely jellemzőjét használhatjuk, figyelmen kívül hagyva az objektumosztályokat. Természetesen ezzel rontjuk a más LDAP-kiszolgálókkal való együttműködés esélyét, sőt bizonyos alkalmazások működését meg is akadályozhatjuk, arról nem is szólva, hogy megcsúfoljuk az LDAP RFC-ket. Nem csoda, hogy a szakértők általában elutasítják a sémaellenőrzés kikapcsolásának ezt a módját.
Talán nincs szükség arra, hogy az 1. kódrészlet minden sorát pontosan elmagyarázzam, a jellemzők jelentése könnyen megérthető. Annyit azért megemlítenék, hogy bár nem kell az összes valaha használni kívánt jellemzőnek értéket adni, némelyik beállítása mégis kötelező – ezeket a megfelelő objektumosztályok megadásában `MUST()`, `MUST()` állításokban találjuk meg. Minden általunk megadott jellemzőnek legalább egy, a rekordban megadott objektumosztály `MUST()` vagy `MAY()` állításában szerepelnie kell, bizonyos jellemzők – mint a `cn` – pedig egy-egy rekordon belül többször is megadhatók.
Az 1. listában látható rekordot az `ldapadd` paranccsal adhatjuk hozzá az adatbázishoz:

```
$ ldapadd -x -D "cn=ldapproba,dc=proba,
➥dc=org" -W -f ./wong.ldif
```

Az előző alkalommal is hasonló módon alkalmaztuk az `ldapadd` parancsot. A parancs használatának módjáról az ldapadd(1) súgóoldalon találni részletes leírást.
Ha az LDIF fájlban megadott objektumosztályok által megkövetelt jellemzők mindegyikét megadtuk, ha a kiválasztott jellemzőket kivétel nélkül támogatják ezek az objektumosztályok, és ha az erre utaló parancs megjelenésekor a helyes LDAP `bind` jelszót adjuk meg, akkor a rekord bekerül az adatbázisba. Ha valamelyik feltétel nem teljesül, akkor a művelet végrehajtása sikertelen lesz, az `ldapadd` pedig kiírja, hogy milyen hiba merült fel. Vagyis: hibáinkból tanulva, próbálgatással is összeüthetjük a megfelelő formátumú rekordot. Ha egyszer már sikerült eredményre jutni, a többi rekordnál is ugyanazt a formátumot tudjuk használni, nem kell újra megszenvednünk a sémával.
Egy apróságra azért felhívnám a figyelmet. Ha az LDIF fájlunk több rekordot is tartalmaz, ami megengedett, és az LDAP-kiszolgáló hibát észlel, akkor abbahagyja a fájl feldolgozását, és a hibás után következő rekordok beolvasását meg sem kísérli. Az első néhány felhasználó hozzáadásakor tehát, amíg a rekordformátum valóban véglegessé nem válik, inkább csak egy-egy rekordot tartalmazó LDIF állományokkal érdemes dolgozni.
A rekordok kézi létrehozása kicsit unalmas, de legalább módot ad egy kis elemzésre és ismerkedésre. Ez a lehetőség különösen az LDAP-adatbázis felépítésének kezdetekor fontos.
Ha már van néhány felhasználórekordunk, akkor valamilyen grafikus felületet nyújtó eszköz, például az LDAP Browser/Editor (➲ http://www.iit.edu/~gawojar/ldap) vagy a `gq` (a legtöbb Linux-terjesztésben megtalálható) segítségével is folytathatjuk munkánkat. Ha például a `gq`-ban rákattintunk valamelyik rekordra, akkor egy menü bukkan fel, amely egy ***New Use current entry*** (Új Pillanatnyi bejegyzés használata) parancsot is tartalmaz, ezzel a kijelölt rekordból készíthetünk másolatot. Így nyilván gyorsabban és egyszerűbben dolgozhatunk, mintha mindent begépelnénk egy LDIF fájlba.

Jelszavak létrehozása

Az 1. kódrészlet kapcsán már utaltam arra, hogy az LDIF fájlokban általában nem adjuk meg a felhasználók jelszavait. Erre a célra egy külön eljárás használható, amely az `ldappasswd` parancsban „testesül meg”. Használata alapvetően az `ldapadd` parancséhoz hasonló:

```
$ ldappasswd -S -x -D
➥"cn=rendszergazda,dc=upstreamsolutions,
➥dc=com" -W "cn=Phil
➥Lesh,dc=upstreamsolutions,dc=com"
```

Az `ldappasswd` parancs használatához nem kell a héjmunkamenetbe bejelentkeznünk az LDAP-kiszolgálón, elég, ha a `-H` kapcsolóval megadjuk a távoli LDAP-kiszolgáló URL-jét:

```
$ ldappasswd -S -x -H
➥ldaps://ldap.upstreamsolutions.com -D
➥"cn=rendszergazda,dc=upstreamsolutions,
➥dc=com" -W "cn=Phil
➥Lesh,dc=upstreamsolutions,dc=com"
```

A kapcsoló az `ldapadd` paranccsal is használható.
A fenti példában szereplő ***ldaps://*** URL megadása kötelező.
A `-x` kapcsolóval egyszerű, nyílt szöveggel történő hitelesítést választottam, így szükségszerűen TLS titkosítással védett kapcsolatot létesítek a kiszolgálóval. A múltkor már megmutattam, hogyan lehet TLS-kapcsolatok fogadására beállítani az LDAP-kiszolgálót.
Nehéz lenne elhallgatnom azt a tényt, hogy LDAP használatakor a felhasználók jelszavainak a kezelése okozza a legtöbb gondot. Ha viszont a felhasználók hozzáférést kapnak az `ldappasswd` parancshoz, akkor helyi ***/etc/ldap.conf*** fájlokkal és parancsfájlokkal vagy megfelelő felületet biztosító programokkal egyszerűen megváltoztathatják a saját jelszavukat.
A más operációs rendszert futtató felhasználók jelszavait azonban központilag kell kezelni, és – hacsak nem telepítünk LDAP ügyfélprogramot az ő gépükre is – a felhasználók minden jelszóváltoztatás alkalmával kénytelenek felvenni a kapcsolatot a rendszergazdával. A Microsoft Windows alapú munkaállomásokhoz a Samba úgy is beállítható, hogy a felhasználók számára lehetővé tegye LDAP-jelszavuknak a Windows beépített jelszókezelőjével történő megváltoztatását (lásd az „OpenLDAP mindenütt” című írást a Linuxvilág 2003. januári számában).

Hozzáférés-vezérlés

Gyakorlatilag minden olyan tennivalót átvettünk, vagy legalábbis érintőlegesen megemlítettünk, amelyet egy OpenLDAP alapú LDAP-kiszolgáló üzembe helyezéséhez el kell végezni (kivéve a különféle kiszolgálóalkalmazások LDAP alapú hitelesítésre való átállításának cseppet sem lebecsülendő feladatát). Ha erőteljes biztonságra törekszünk, márpedig ha az egészbe belekezdtünk, csakis ez lehet a cél, akkor még egy dologról szót kell ejtenünk: az OpenLDAP hozzáférés-vezérlési listáiról (ACL).
Ahogy a `slapd` démonra vonatkozó beállítások többségének, úgy az ACL-eknek a megadása is a ***/etc/openldap/slapd.conf*** fájlban történik. Az LDAP kapcsán talán nem meglepő, hogy az ACL-ek kezelése (is) okozhat némi fejtörést, és a kívánt eredmény eléréséhez bizony szükség lesz némi kísérletezésre. A *2. lista* az ACL-ek megadására mutat példát.
Az ACL-ek részletes ismertetése a slapd.conf(5) súgóoldalon található meg, de működésük alapjait a 2. kódrészlet alapján is fel lehet ismerni. Minden védeni kívánt LDAP-elemhez meg kell adnunk, hogy pontosan kik kapnak hozzáférést, illetve ez milyen szintű legyen. Egyébként a teljes ACL-t egyetlen sorban meg lehetne adni, de a hagyomány az, hogy minden "***by...***" állítást külön sorba írunk. A `slapd` elég okos ahhoz, hogy tudja, az "***access to***" karakterlánc a következő ACL kezdetét jelzi. Helyszűke miatt részletesen nem ismertethetem az ACL-ek

2. lista ACL-ek a /etc/slapd.conf fájlban

```
access to attrs=userPassword
  by dn="cn=ldapproba,dc=proba,dc=org" write
  by self write
  by * compare

access to *
  by dn="cn=ldapproba,dc=proba,dc=org" write
  by users read
  by * auth
```

írásmódját, így csak néhány dolgot emelnék ki. Először is, az ACL-ek feldolgozása fentről lefelé történik, és az első találat jut érvényre, mintha szűrők kupacáról lenne szó. Fontos tehát, hogy a különleges ACL-eket az általánosabbak fölé helyezzük. A 2. listában például egy olyan ACL-t láthatunk, amely a `userPassword` jellemző elérését korlátozza, ezt pedig egy a teljes LDAP-adatbázisra vonatkozó szabály követi.
A `userPassword` ACL-t előre helyezve engedélyezhetjük a felhasználóknak saját jelszavuk módosítását (`access to attrs=userPassword by self write`), de ez a lehetőség kivételt képez ahhoz képest, hogy a felhasználók bármit olvashatnak (`access to * by users read`).
Ugyancsak hangsúlyoznom kell, hogy a hozzáférési szintek hierarchiába szerveződnek. A lehetséges szintek a `none`, az `auth`, a `compare`, a `search`, a `read` és a `write`, ahol a `none` a legalacsonyabb szint, a `write` pedig a legmagasabb, és mindegyik szint magába foglalja a nála alacsonyabbak által megadott jogokat is. Az első egyezés érvényre jutása és a hozzáférési szintek egymást való tartalmazása az a két jellegzetesség, amelyet az ACL-ek kezelése kapcsán meg kell érteni, már csak azért is, mert gondosan ügyelni kell arra, hogy az ACL-ek ne adjanak a kívántnál bővebb jogokat, de a munkához szükséges hozzáféréseket biztosítsák.

Összegzés

Az LDAP az általam az utóbbi időkben látott egyik legbonyolultabb megoldás. Ha saját elvárásaink szerint szeretnénk munkára bírni, akkor fel kell készülnünk arra, hogy rengeteg próbálgatásra lesz szükségünk, naplókban kell hosszasan turkálnunk, és mind az LDAP-kiszolgáló, mind a segítségével hitelesítést végző alkalmazások oldalán idegölő finomhangolást kell végeznünk. Ha azonban figyelembe vesszük, hogy milyen rugalmas, nagy teljesítményű és széles körben támogatott hitelesítő és címtárkezelő eszközt kapunk, máris enyhülnek az álmatlan éjszakák kínjai. Remélem, hogy írásaim segítségével sikerül ezt a munkát véghez vinni, vagy legalábbis közelebb jutni a kitűzött célhoz.

A cikkhez tarozó Kapcsolódó címek megtalálhatóak az 53. CD Magazin/LDAP könyvtárában.

Linux Journal 2003. szeptember, 113. szám

Mick Bauer (mick@visi.com)
Biztonsági szakember, a Linux Journal biztonsági témákkal foglalkozó szerkesztője, biztonsági tanácsadó a Minnesota állambeli Minneapolisban található Upstream Solutions LLC Inc.-nél. Mick szabadidejében a gyermekeivel fogócskázik vagy zenél.