

Hitelesítés LDAP használatával (2. rész)

A címtárkiszolgáló működik, tehát határozzuk meg a titkosítási beállításokat, majd adjuk hozzá a rendszerhez a felhasználókat.

A múlt hónapban már elvégeztük az OpenLDAP-kiszolgáló üzembe helyezésével járó munka egy részét. Telepítettük az alapot, vagyis a kiszolgálót, illetve a szükséges OpenLDAP-ügyfélcsomagokat, majd megadtunk bizonyos alapvető beállításokat a `/etc/openldap/slapd.conf` fájlban (a `slapd` az OpenLDAP kiszolgálódémona). Ez alkalommal a TLS alapú titkosítást állítjuk be, elindítjuk a demont, és megkezdjük egy LDAP-adatbázis felépítését. Teljesen ugyan nem fogjuk befejezni a hitelesítő kiszolgálót, ám nagyon közel kerülünk ehhez az állapothoz. A jövő hónapban megjelenő cikkben – ami egyben a három részből álló sorozat utolsó tagja lesz – a maradék munkát is letudjuk.

LDAP-tranzakciók TLS alapon

Az OpenLDAP adatseréi alapesetben nyílt szövegek használatával folynak a hálózaton keresztül. Ha az OpenLDAP például központi telefonkönyv-kiszolgálóként teljesít szolgálatot egy megbízható hálózaton, akkor ezzel nincs is semmi gond. Ha azonban a hálózat megbízhatóságával nem foglalkozva felhasználók hitelesítésére használjuk, akkor jobb, ha a hallgatózók ellen titkosítással védjük az LDAP párbeszédet, illetve egyben a felhasználók jelszavait.

Az LDAP v.3 protokoll, amelynek támogatása az OpenLDAP 2.0-s változatában jelent meg, képes a Transport Layer Security (átvitelirétegbeli biztonság, TLS) alapú védelemre – ezt a megoldást a böngészők és a levelezőprogramok is használják. A TLS az SSL (Secure Sockets Layer) utódja. Ha élvezni akarjuk a TLS nyújtotta előnyöket, az LDAP-kiszolgálón mindössze létre kell hoznunk egy kiszolgálótanúsítványt, hozzá kell adnunk néhány sort a `/etc/openldap/slapd.conf` fájlhoz, majd egy picit meg kell piszkálnunk a `slapd` indítási beállításait. A kiszolgálótanúsítvány létrehozásához OpenSSL-re van szükség. Ez valószínűleg már megtalálható a rendszeren, a bináris OpenLDAP-csomagok ugyanis az OpenSSL-től függenek. Az, hogy LDAP-tanúsítványként pontosan milyen tanúsítványt érdemes használni, bizony, fogas kérdés. Olyan tanúsítványra van szüksége a kiszolgálónak, amelyet valamelyik hitelesítő szervezet (certificate authority, CA, mint például a VeriSign vagy a magyar NetLock) írt alá? Vagy a másik oldalról megközelítve: az LDAP-ügyfeleknek külső fél által ellenőrizhető tanúsítványt kell látniuk, amikor csatlakoznak a kiszolgálóhoz? Esetleg a saját szervezetünk ön maga hitelesítő szervezete lesz? Az utóbbi esetben helyi CA-ként is szolgál majd az LDAP-kiszolgáló, ami gondoskodik a saját tanúsítványának, valamint a többi állomás és felhasználó tanúsítványainak kibocsátásáról és aláírásáról? Ha valamelyik kérdésre igen a válasz, akkor egy kicsit több dolgot kell elintézni, mint amennyit itt leírhatnék. Legyen elég annyi, hogy a `slapd` által használt tanúsítványhoz nem tartozhat jelszó – vagyis a kulcsa nem lehet DES kódolású –, így ön-aláírt tanúsítvány, hiába CA-tanúsítvány, valójában nem használható CA-tanúsítványként más tanúsítványok aláírására. Ha azt akarjuk, hogy az LDAP-kiszolgáló valódi CA-ként működjön, akkor két kulcsot kell létrehozunk: egy jelszóval

védett CA-kulcsot és egy jelszó nélkül használható `slapd`-kulcsot. A témával *Vincent Danen* Using OpenLDAP for Authentication című cikkében foglalkozik bővebben (☞ <http://www.mandrakesecure.net/en/docs/ldap-auth.php>). Az esetek túlnyomó részében elég egy saját TLS-tanúsítványt létrehozni, amelyet kizárólag a `slapd` használ. Ha CA-t nem akarunk létrehozni, illetve az LDAP-ügyfelek sem akarják külső fél segítségével ellenőrizni a tanúsítvány hitelességét, akkor a tanúsítványt az alábbiak szerint hozhatjuk létre:

```
bash-5$ openssl req -new -x509 -nodes -out
↳slapdtanositvany.pem -keyout slapdkulcs.pem
↳-days 365
Using configuration from
↳/usr/share/ssl/openssl.cnf
Generating a 1024 bit RSA private key
.....+++++
.....+++++
writing new private key to 'slapdkulcs.pem'
```

A fenti paranccsal arra utasítottam az OpenSSL-t, hogy hozzon létre egy új, jelszavas védelem nélküli X.509 tanúsítványt, majd ezt (a nyilvános kulcsot) írja ki a pillanatnyi munkakönyvtárba, a `slapdtanositvany.pem` nevű fájlba, a titkos kulcsot pedig a `slapdkulcs.pem` nevű állományba mentse. Élettartamként 365 napot adtam meg.

A parancs kiadása után meg kell adnunk a tanúsítványhoz és a kulcshoz tartozó megkülönböztető neveket. Az OpenLDAP esetében a legfontosabb mező a közös név (CN, common name). Ennek egyeznie kell az LDAP-kiszolgáló DNS-nevével. A tanúsítványhoz hozzárendelt névként az LDAP-ügyfelek ezt a nevet fogják látni. Ha például az LDAP-kiszolgáló IP-címéhez a `bonzo.lamemoviesfromthepast.com` név tartozik, ám a kiszolgáló tanúsítványban megadott CN értéke `bonzo.lm.com`, akkor az LDAP-ügyfelek vissza fogják utasítani a tanúsítványt, és így képtelenek lesznek TLS-kapcsolatokat teremteni (az eredményt megjósolni nem lehet, az az adott ügyfélprogramtól függ). Ha megvan a tanúsítvány és a kulcs, a fájlokat másoljuk a `/etc/openldap` könyvtárba, feltéve, hogy eddig nem ezt használtuk munkakönyvtárként. Mindkét fájl tulajdonosa az `ldap` legyen, illetve az a felhasználó, akinek a neve alatt a `slapd` fut. Red Hat és SuSE alatt ez az `ldap`. A kulcsfájllra a legszigorúbb engedélyeket kell kiadni, például:

```
--r-----
```

A tanúsítványt tároló fájl tulajdonképpen bárki olvashatja, hiszen nyilvános kulcs található benne.

Arra is van lehetőség, hogy a `-out` és a `-keyout` kapcsoló után ugyanazt a fájlnevet adjuk meg, ekkor a tanúsítvány és a kulcs ugyanabba az állományba kerül. Ennek akkor van értelme, ha a tanúsítványt nem akarjuk megosztani. A két különálló fájl használata ugyanakkor lehetővé teszi a kiszolgáló

A /etc/openldap/slapd.conf fájl testreszabott része

```
database            ldbm
suffix              "dc=proba,dc=org"
rootdn
"cn=ldapproba,dc=proba,dc=org"
rootpw
{SSHA}zRsCkoVvVDXObE3ewn19/Imf3yDoH9
directory           /var/lib/ldap
TLSCipherSuite      HIGH:MEDIUM:+SSLv2
TLSCertificateFile  /etc/openldap/slaptanositvany.pem
TLSCertificateKeyFile
/etc/openldap/slapdkulcs.pem
```

tanúsítványának terjesztését, miközben a titkos kulcsot valóban titokként kezelhetjük.

Természetesen nem elég a tanúsítvány- és a kulcsfájl bemásolása a megadott könyvtárba, a `slapd`-t is utasítanunk kell a használatukra. A `slapd` beállításainak túlnyomó részéhez hasonlóan erről is a `/etc/openldap/slapd.conf` fájlban rendelkezhetünk. A `listát` a múlt hónapban már megismert, azóta tán már el is felejtett `slapd.conf` fájl bejegyzéseit tartalmazza – ezekhez három új sor társul: `TLSCipherSuite`,

`TLSCertificateFile` és `TLSCertificateKeyFile`.

A `TLSCipherSuite` azokat az OpenSSL titkosítási eljárásokat sorolja fel – a leginkább kívánatosul kezdve –, amelyek közül a `slapd` a TLS-kapcsolatok egyeztetésekor választhat. Azt, hogy a helyi gépre telepített OpenSSL-példány mely titkosítási eljárásokat támogatja, a következő paranccsal állapíthatjuk meg:

```
openssl ciphers -v ALL
```

A titkosítási eljárások felsorolása mellett az OpenSSL által támogatott helyettesítő szavakat is használhatjuk, így egy-egy szóval több eljárást is kiválaszthatunk. Az 1. kódrészletben például a `TLSCipherSuite` értéke `HIGH:MEDIUM:+SSLv2`. A `HIGH`, a `MEDIUM` és a `+SSLv2` kivétel nélkül helyettesítő szavak.

A `HIGH` jelentése: „minden olyan titkosítási eljárás, amelyik 128 bitesnél nagyobb kulcshosszal dolgozik”; a `MEDIUM` a „minden 128 bites kulcsot használó eljárás” rövidítéseként fogható fel, a `+SSLv2` pedig a „minden az SSL-protokollban megadott eljárás, tekintet nélkül a kulcshosszra” szinonimájaként kezelendő. Az OpenSSL titkosítási eljárások részletesebb ismertetését, illetve a támogatott helyettesítő szavakat a `ciphers(1)` sűgőoldalon lehet megtalálni.

A `TLSCertificateFile` és a `TLSCertificateKeyFile` kapcsoló jelentése kézenfekvő: a tanúsítványt és a titkos kulcsot tartalmazó fájl nevét adják meg. Ha a tanúsítványt és a kulcsot ugyanabba a fájlba helyeztük el, akkor mindkét kapcsolónak ugyanazt az értéket kell adni.

A slapd indítási kapcsolói

A kiszolgáló oldalán minden szükséges lépést elvégeztünk annak érdekében, hogy a TLS titkosítás működjön. Most már csak egy kérdésben kell döntenünk. A TLS használata az összes LDAP-kérés esetében kötelező legyen, vagy mint választható lehetőséget kínáljuk fel?

Alapesetben a `slapd` a 389-es TCP-kapun fogadja az LDAP-kéréseket, legyenek azok akár nyílt szövegben érkezők, akár titkosítottak. Ha az LDAP-t hitelesítési célokra akarjuk használni,

valószínűleg érdemes kötelezővé tenni a TLS használatát.

Ebben az esetben jobb, ha a `slapd` a nyílt szöveggként érkező kéréseket csak a helyi hurokfelület 389-es TCP-kapuján keresztül fogadja, a TLS alapúakat pedig az összes helyi cím 636-os TCP-kapuján várja – ez egyébként az `ldaps` szabványos kapuja. Mindezt a `slapd -h` indítási kapcsolójával szabályozhatjuk, amely a `slapd` által a kérések fogadására használt URL megadására szolgál. A `slapd -h ldap://127.0.0.1/ldaps:///` parancs hatására például a `slapd` a helyi hurokfelületen (127.0.0.1) az alapértelmezett kapun (TCP 389) át fogadja az `ldap`-kapcsolatokat, a többi helyi címen pedig az alapértelmezett `ldaps` kapun (TCP 636) keresztül várja az `ldaps` kéréseket. Ha Red Hat 7.3 vagy újabb rendszert használunk, ez egyben az alapértelmezett beállítás is. A `/etc/init.d/ldap` a `/etc/openldap/slapd.conf` fájlban keresi a TLS-beállításokat, és ha talál ilyeneket, akkor a `-h` kapcsolót pontosan az iménti példa szerint alkalmazza. Ha SuSE 8.1 vagy újabb terjesztést futtatunk, akkor ezt a viselkedést a `/etc/sysconfig/openldap` fájl módosításával érhetjük el. Az `OPENLDAP_START_LDAPS` beállításnak *yes*, a `/etc/init.d/openldap` fájlban található `SLAPD_URLS` beállításnak pedig `ldap://127.0.0.1` értéket kell adnunk. A változó megadása a parancsfájl elején található, alapértelmezett értéke a `ldap:///`. Más Linux-terjesztéseknél ettől eltérő módon történhet az indítási kapcsolók, köztük a `-h` átadása a `slapd` számára, de a fentiek alapján remélhetőleg nem lesz túl nehéz a kívánt kapuk beállítása.

Próba

Nos, valóban működik TLS alapú LDAP-kiszolgálónk? Egy gyors helyi próbával megkapjuk a választ. Először is indítsuk el az LDAP-t:

```
bash-$ /etc/init.d/ldap start
```

Ezután az `ldapsearch` parancs segítségével végezzünk el egy egyszerű lekérdezést a hurokfelületen keresztül:

```
bash-$ ldapsearch -x -H ldaps://localhost/
-b 'dc=proba,dc=org' '(objectclass=*)'
```

Természetesen a saját LDAP-kiszolgálónk neve nem `dc=proba,dc=org` lesz. Ha gondoljuk, ezt a parancsot egy távoli gépen is kiadhatjuk, ekkor a `-h` kapcsolónál a `localhost` helyett az LDAP-kiszolgáló nevét vagy IP-címét kell megadnunk. Ha az LDAP-kiszolgáló válaszul kiírja az – egyelőre üres – LDAP-adatbázis tartalmát, majd a `0 Success` felirat jelenik meg, akkor a próba sikeres volt.

Ha érvénytelen tanúsítványra utaló hibaüzenetet kapunk, akkor próbáljuk hozzáadni az alábbi sort az ügyfélgép `/etc/openldap/ldap.conf` állományához:

```
TLS_REQCERT          allow
```

Ezzel engedélyezzük az OpenLDAP vagy az OpenLDAP alapú ügyfélprogram (például `gq`) számára, hogy önaláírt kiszolgáló tanúsítványokat is elfogadjon.

LDAP-séma

Pillanatok múlva megkezdhetjük az LDAP adatbázis feltöltését. A megfelelő eszközök segítségével – ilyen a `gq` és az `ldapbrowser` – jelentősen csökkenthető az LDAP-adatok bevitele és a felügyelete miatti álmatlan éjszakák száma. Csakhogy ahhoz, hogy ezeket az eszközöket használni tudjuk, először meg kell alkotnunk a megfelelő LDAP-sémát, és a történet itt kezd kacifántos lenni.

Esetünkben két LDAP-adattípussal érdemes foglalkozni, az egyik az attribútum avagy jellemző, a másik az objektumosztály. A jellemzők alkotják a rekordokat. Ilyen jellemző például a felhasználók levélcíme, beceneve, telefonszáma. Az LDAP-adatbázis tetszőleges számú jellemzőt kezelhet, akár saját jellemzőket is kitalálhatunk. Ahhoz azonban, hogy egy rekord egy adott jellemzőt tartalmazhasson, a rekordot a megfelelő objektumosztállyal össze kell rendelni. Az objektumosztály a felépítendő rekord típusát írja le. Megadja, hogy az egyes rekordok esetében mely jellemzők megadása kötelező, és melyek hagyhatók el. Ennek alapján azt is gondolhatjuk, hogy könnyű dolgunk lesz, hiszen csak ki kell választanunk azt az objektumosztályt, amelyik az általunk fontosnak tartott jellemzőket tartalmazza, majd az összes felhasználórekordot ehhez az osztályhoz rendeljük hozzá. Az élet azonban sajnos nem ilyen egyszerű. A gyakorlatban nagy valószínűséggel különféle objektumosztályok jellemzőit akarjuk majd használni. Semmi gond, véljük, minden felhasználórekordhoz több objektumosztályt is hozzárendelünk, és tetszésünkre csemegézünk majd a jellemzők közül. Ebben is van valami, de a dolog nem tudható le ennyivel. A szükséges jellemzőket megadó objektumosztályok jó eséllyel különféle sémafájlokba vannak szétszórva (a sémafájlok szöveges állományok, ezek jellemzőket és rájuk hivatkozó objektumosztályokat tartalmaznak). Mielőtt tehát megkezdénénk saját, jó néhány objektumosztály-hivatkozást és még több jellemzőt tartalmazó rekordjaink létrehozását, először ellenőriznünk kell, hogy a `/etc/openldap/slapd.conf` fájl az összes szükséges sémafájltra – ezek általában a `/etc/openldap/schema` könyvtárban vannak – tartalmaz-e hivatkozást. Tegyük fel például, hogy LDAP-kiszolgálókat azonosítási célokra akarjuk használni, ezért a `userId` és a `userPassword` jellemzőkről semmilyen körülmények közt nem szeretnénk lemondani. A `grep` segítségével hamar kideríthetjük, hogy a `/etc/openldap/schema` könyvtár fájljai közül az `uid` az `inetOrgPerson.schema` fájl `MAY` listájában (a megengedett jellemzők közt), az `inetOrgPerson` objektumosztály alatt található. Ennek két vonzata van. Az első az, hogy a `/etc/openldap/slapd.conf` fájlban tartalmaznia kell az alábbi sort:

```
include /etc/openldap/schema/inetOrgPerson.schema
```

A második az, hogy amikor felhasználórekordot hozunk létre, akkor meg kell vizsgálni, hogy a `objectclass`: `inetOrgPerson` létezik-e.

Felhasználórekordok létrehozása és hozzáadása

A múlt hónapban már említettem a `gq-t`, amely számos terjesztésben megtalálható. Ugyancsak kiváló program az `ldapbrowser`, amely a `http://www.iit.edu/~gawojar/ldap` címen érhető el. Kezdeként azonban előfordulhat, hogy például a szervezetünk bejegyzését kézzel hozzuk létre. Ehhez készítenünk kell egy `ldif` fájlt, majd az `ldapadd` paranccsal be kell emelnünk a tartalmát az adatbázisba. Az `ldif` fájl egy olyan szöveges állomány, amely jellemző/objektumosztály megadásokat tartalmaz (soronként egyet), például:

```
dn: dc=proba,dc=org
objectclass: top
objectclass: organization
o: proba kiszolgalo
```

A fentiek szerint a `proba.org` szervezetet határozzuk meg. Megadjuk megkülönböztető nevét, összerendeljük a `top` (minden rekordnál kötelező) és az `organization` objektumosztállyal, majd megadjuk a szervezet nevét (`proba kiszolgalo`), ennél a két objektumosztálynál ez az egyetlen kötelező jellemző. A rekordot az alábbi paranccsal írhatjuk be az adatbázisba:

```
bash-$ ldapadd -x -H ldaps://localhost/
-D "cn=ldapproba,dc=proba,dc=org"
-W -f proba_adatok.ldif
```

Mint a legtöbb OpenLDAP parancsnál, a `-x` egyszerű jelszavas azonosítást ír elő, a `-H` az LDAP kiszolgáló URL-jét, a `-D` pedig a rendszergazdai fiók megkülönböztető nevét adja meg, míg a `-W` a rendszergazdai jelszó bekérését váltja ki. A `-f` kapcsolót az `ldif` fájl elérési útja követi.

Alakul? Lehet, hogy egy kicsit sok dolgot kellett most megemléstenni, de vigasztaljon mindenkit az a tudat, hogy az LDAP-kiszolgáló kis híján készen áll.

Linux Journal 2003. augusztus, 112. 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.

