Központosított hitelesítés és vállalati címtár megvalósítása (1. rész)

Ti Leggett sorozatában arról lesz szó, hogy hogyan lehet biztonságos vállalati címtárat létrehozni, amely támogatja az egyszeri bejelentkezést, és amely felhasználók ezreit képes kiszolgálni.

Vállalati címtárat szeretnénk tehát, de nincs vállalati költségvetésünk. Az egyszeri bejelentkezés előnyeit szeretnénk élvezni, mely mind az adminisztrátor, mind a felhasználók életét megkönnyíti. Ha ez a cél, és ráadásként elfogadnak egy biztonságos és egységesített hitelesítési és személyazonosság-nyilvántartási rendszert, akkor érdemes kitartóan végigolvasni e sorokat. Elindítom az olvasót a rendszergazda-mennyországba vezető úton. Ebben a cikksorozatban megmutatom, hogyan lehet építkezni a már előállított és helyretett alapelemekből, hogyan lehet újabbakat is beépíteni, és miként lehet ezek együttesét munkára bírni. A hitelesítést intéző kiszolgálóktól kezdve a levélkézbesítésig és kliensgépek integrálásáig (beleértve a ***Microsoft Windows*** vagy ***OS X*** operációs rendszert futtató gépekig) mindent végiggondolunk. Sok mindenről kell szót ejtenünk, úgyhogy kezdjünk is bele!

A korábbi építőkockák felhasználása

Gentoo Linuxon futtatott ***MIT Kerberos V v1.4.1***-et használunk hitelesítésre és ***OpenLDAP v2.1.30***-at címtárként, azaz személyazonosság-kezelésre. Három kiszolgálónk lesz: ***kdc.pelda.com***, ***ldap.pelda.com*** és ***mail.pelda.com***. (A beszédes nevek az alábbiakra utalnak: ***KDC=Key Distribution Center, Kulcselosztó Központ; LDAP=Lightweight Directory Access Protocol,*** címtár szolgáltatások elérését szabályozó protokoll) Mielőtt továbblépnénk, érdemes elolvasni a ***Linuxvilágból*** a ***„Központosított hitelesítés Kerberos 5-tel”*** cikksorozat első részét ➲ ***linuxvilag.hu/node/3002370***, valamint az ***„OpenLDAP mindenütt”*** című cikket ➲ ***linuxvilag.hu/node/3001551***, illetve ***Jászberényi József*** esettanulmányát 2006 szeptemberében és októberében. (Lásd az on-line forrásokat, valamint a ***Google-beli „LDAP OpenSUSE” keresőkifejezés is jó magyar bevezetőt ad az LDAP témához – a ford.***)
Onnan lépünk tovább, ahová ezek az írások eljutottak. Azt érdemes még szem előtt tartani, hogy a mi ***Kerberos*** tartományunk (***realm***) a ***CI.PELDA.COM***, így bázis ***DN***-ünk az ***o=ci***, ***dc=pelda***, ***dc=com*** lesz. (***DN = Distinguished Name***, megkülönböztető név; ***o = organization***, szervezet; ***dc = domain component***, tartománykomponens)
A cikkben hivatkozott összes konfigurációs fájl megtalálható a megjelölt on-line források között.

SSL Tanúsítvány Hatóság (Certificate Authority, CA) létrehozása

A továbbiak megértéséhez ezt a részt nem kötelező elolvasni, de olyan hálózatok építői számára, akik több szerveren is ***SSL***-t használnak, melegen ajánlott. Bár minden kiszolgáló alá tudná írni a maga számára a tanúsítványt, így elveszne valami az egységességből és a saját ***CA*** futtatásával járó sokféle lehetőségből. Az ***OpenSSL*** részletei iránt érdeklődőknek jó szívvel tudom ajánlani a ***Network Security with OpenSSL*** című könyvet (***John Viega, Matt Meisser, Pravir Chandra – O'Reilly***).
Legyen a ***/etc/ssl/pelda.com*** könyvtár az alapkönyvtár, ahol az összes aláírt tanúsítványt, visszavont tanúsítványlistát (***certificate revocation list, CRL***) és azonosítói információt tartjuk.
Ha e könyvtár elkészült, hozzuk létre benne a ***certs, crl, newcerts*** és ***private*** alkönyvtárakat, valamint ugyanitt egy üres ***/etc/ssl/pelda.com/index.txt*** fájlt. Írjuk egy „01”-t a ***/etc/ssl/pelda.com/serial*** nevű fájlba.
Ez utóbbiakat megtehetjük pl. az alábbi módon:

```
# touch /etc/ssl/
# pelda.com/index.txt
# echo '01' > /etc/ssl/
# pelda.com/serial
```

Végül hozzunk létre a ***CA*** számára egy ***OpenSSL*** konfigurációs fájlt ***/etc/ssl/pelda.com/ca-ssl.cnf*** néven.
Jelentkezzünk be olyan felhasználói néven, aki a ***/etc/ssl/pelda.com*** könyvtár rekurzív tulajdonosa (valószínűleg a ***root*** felhasználó). Az ő nevében az alábbiakat kell tennünk egy ön-aláírt ***CA*** tanúsítvány létrehozásához:

```
# export OPENSSL_CONF=
# /etc/ssl/pelda.com/
# ca-ssl.cnf
# openssl req -x509 -days 3650
# -newkey rsa \
 -out /etc/ssl/pelda.com/
 ➥ci-cert.pem -outform PEM
```

```
# cp /etc/ssl/pelda.com/
# ci-cert.pem /etc/ssl/certs
# /usr/bin/c_rehash
# /etc/ssl/certs
```

Az ***openssl req*** parancs részleteire vonatkozóan eligazítást ad a ***req(1)*** kézikönyv (man) oldal.
Fontos, hogy a ***CA*** kulcs jelszava (***passphrase***) igen biztonságos helyen legyen, mert ha a ***CA*** titkos kulcsa kitudódik, megbízhatatlanná válnak az általa aláírt tanúsítványok. Az is fontos, hogy a ***CA*** számítógép maga, illetve az elérése is biztonságos legyen. Ennek a biztonságnak a szintje az adminisztrátortól és az általa képviselt igényektől függ, de mihelyt egy illetéktelen felhasználó fizikailag vagy a hálózaton keresztül hozzá tud férni e géphez, már meg is szerezte a ***CA*** titkos kulcsát. Ahogy már fentebb említettem, a ***CA*** titkos kulcsának veszélyeztetése lerombolja az egész bizalmi láncot, bizonytalanná és megbízhatatlanná válik az összes aláírt tanúsítvány. Egyesek szerint az a legjobb megoldás, ha a ***CA*** gép fizikailag el van különítve mindenféle hálózattól.
A tanúsítványok aláírására úgy kerülhet sor egy ilyen környezetben, hogy ***Tanúsítványregisztrációs Központok (Registration Authorities, RA)*** fogadják a beérkező ***Tanúsítványaláírási Kérelmeket (Certificate Signing Request, CSR)***. Itt ezeket a ***CSR*** kéréseket egy megbízható adathordozón átviszik a CA gépre, ahol sor kerül az aláírásra és a tanúsítványoknak az adathordozóra történő visszaírására, amit a ***Tanúsítványregisztrációs Központtól*** a végfelhasználó átvehet.
Amennyiben erre van szükségünk, akkor az ***OpenCA*** projekt ilyen biztonsági szintet céloz meg; támogatja az aláírt tanúsítványok ***LDAP***-ben való eltárolását is.
Elkészítettük immár a ***CA*** számára az ***OpenSSL*** konfigurációs fájlunkat, de ez csak egyetlen tanúsítvány igénylésére és aláírására alkalmas. Elő kell állítanunk még egy olyan konfigurációs fájlt is, amelyet mostantól normál gazdagép- és felhasználói tanúsítványok igénylésére is lehet használni. Erre szolgál a ***/etc/ssl/pelda.com/ssl.cnf***. A kliensgépek konfigurálása a ***CA***-nál egy kissé összetettebb feladat, mivel többféle tanúsítvány-változatot kell kell kezelnie.

Most, hogy már van egy kliens konfigurációs fájlunk, generáljunk egy gazdagép tanúsítványt is az ***LDAP*** kiszolgáló számára. A ***CSR (Tanúsítvány-aláírási Kérelem)*** előállítása normál felhasználóként történhet:

```
# export OPENSSL_CONF=
# /etc/ssl/pelda.com/
# ssl.cnf
# openssl req -new -nodes
# -keyout ldap-key.pem \
 -out ldap-req.pem
```

Az ***openssl req*** kapcsolói lényegében ugyanazok, mint amiket a ***CA CSR*** legyártásához használtunk.
Az egyetlen újdonság a `-nodes` opció, ami egy kódolatlan titkos kulcsot gyárt (***man req***).
A nyilvános tanúsítvány elkészítéséhez vezető következő lépés a ***CSR*** aláíratása a ***CA***-val. Ezt ismét ***root***-ként kell megtenni:

```
# export OPENSSL_CONF=
# /etc/ssl/pelda.com/
# ssl.cnf
# openssl ca -policy
# policy_anything -out \
 ldap-ckert.pem -in ldap-
 ➥req.pem
```

Ebben a pillanatban három fájlunk van: az ***ldap-cert.pem***, a nyilvános tanúsítvány ; az ***ldap-key.pem***, a titkos kulcs; valamint az ***ldap-req.pem***, a ***CSR (Tanúsítvány-aláírási Kérelem)***.
Ez utóbbi eldobható, mihelyt a tanúsítványt aláírta a ***CA***. Itt is hangsúlyoznunk kell, mennyire fontos a titkos kulcs védelme, annál is inkább, mert nincs kódolva. Érdemes ezt a ***root*** birtokában tartani és 0400 jogosultsággal ellátni.

Legyen biztonságos az LDAP

Annak ellenére, hogy nincsenek jelszavak az ***LDAP*** címtárban, mégis van itt sok más értékes információ. Vélhetőleg a felhasználók nem örülnének, ha az interneten nyilvánosságra hoznák a telefonszámaikat, e-mail címeiket vagy munkavállalói azonosító számaikat. Ha olvasták az ***„OpenLDAP mindenütt”*** című írást és van egy működő ***LDAP*** kiszolgálójuk, akkor szükség van arra is, hogy biztonságosan történhessen az adatok átvitele és a címtár elérése.

Az adatátvitel biztosításának első lépése az ***OpenSSL*** használata.
Először is másoljuk át az aláírt tanúsítványunkat illetve kulcsunkat a ***/etc/openldap/ssl/slapd-cert.pem*** és a ***/etc/openldap/ssl/slapd-key.pem*** helyekre.A ***slapd.conf***-ban meg kell adnunk öt adatot: ***TLSCipherSuite*** (opcionális), ***TLSCACertificatePath***, ***TLSCertificateFile***, ***TLSCertificateKeyFile*** és ***TLSVerifyClient***. A ***slapd.conf(5)*** kézikönyv oldalak eligazítást adnak ezek mibenlétéről. (***TLS: Transport Layer Security; a protokoll elsődleges célja a titkosság és az adatintegritás biztosítása***)
Most, hogy biztonságosakká tettük a kábeleken áthaladó adatforgalmat, biztosítanunk kell a ***Kerberos KDC*** által használt hitelesítést is.
Az ***OpenLDAP*** „kerberizált”; ***SASL***-t (***„Simple Authentication and Security Layer”; „Egyszerű hitelesítési és biztonsági réteg”***) hitelesítési mechanizmust használ. Először is tudatnunk kell a ***slapd***-vel, hogy hol találja a kulcsokat tartalmazó ***keytab*** fájlt. Ezt a ***/etc/conf.d/slapd*** szerkesztésével tehetjük meg, vagy a ***slapd*** indítása előtt a megfelelő indító szkriptben létrehozott `KRB5_KTNAME` változó segítségével. A ***slapd.conf***-ban meg kell még adni a ***sasl-secprops*** és a ***sasl-regexp*** értékét.
E pillanatban mind a ***TLS***, mind a ***SASL*** mechanizmus használható, de ezek akár nélkülözhetőek is.
Még két opció van a ***slapd.conf***-ban (***security*** és ***allow***), melyek arra használhatóak, hogy megadjuk a biztonsági mechanizmust és a titkosítás erősségét, melyet néhány művelet elvégzése megkövetel. Győződjünk meg arról is, hogy az ***hozzáférés-szabályozó listák (ACL; Access Control List)*** megfelelően be vannak-e állítva. Javasolt kézikönyv-oldalak: ***slapd.access(5)***.

A Kerberos biztonságos átvezetése

Kezdjük azzal, hogy átvezetjük („replikáljuk”) a ***Kerberos*** adatbázisunkat a ***kdc.pelda.com***-ról az ***ldap.pelda.com***-ra. Ha valami miatt a ***kdc.pelda.com*** elromlana, az ***ldap.pelda.com*** át tudja majd venni a szerepét. Emlékeztetnék rá, hogy egy időpontban csak egyetlen ***kadmin*** kiszolgáló lehet a hálózati

tartományunkban. Máskülönben nem lehetne tudni, ki az illetékes az adatbázis-frissítések ügyében.
A *Kerberos* tartalmazza a *kprop* és *kpropd* programokat. Ezek megfelelő módon, biztonságosan el tudják terjeszteni a *Kerberos* adatbázist.
Először is meg kell adnunk a *kpropd*-t, mint ismert szolgáltatást. Írjuk a */etc/services* fájlba:

```
krb5_prop                754/tcp
```

Definiálnunk kell egy *ACL* fájlt is, a */etc/krb5kdc/kpropd.acl*-t, ami felvilágosítja a *kpropd*-t arra vonatkozóan, hogy mely gépek jogosultak a az adatterjesztésre. Igazából csak egy *KDC* főfiókot kell megadnunk a fájlban, de megadható akár az összes *KDC* gép is. Ekkor hiba esetén választhatunk egy új főfiókot, elindíthatjuk rajta a *kadmin* szolgáltatást, és tőle kezdődhet az adatterjesztés a szolgagépek felé.
Ezek után a szolgagépeken megadjuk az *xinetd* szolgáltatás-definíciót a */etc/xinetd.d/kpropd* fájlban; (újra)indítjuk az *xinetd*-t; kiírjuk az adatbázist a *kdc.pelda.com* gépen; és átvisszük a szolgagépekre az alábbi kezdőkonfigurációval:

```
# /usr/sbin/kdb5_util dump
# /etc/krb5kdc/slavedump
# /usr/sbin/kprop -f
# /etc/krb5kdc/slavedump \
 ldap.pelda.com
```

Végül minden szolgagépen létrehozunk egy biztonsági (*stash*) fájlt, mégpedig annak a mesterkulcsnak a segítségével, amit a *kdc.pelda.com* adatbázisának beállításakor használtunk; aztán elindítjuk a *KDC* szolgáltatást:

```
# /usr/sbin/kdb5_util stash
# /etc/init.d/mit-krb5kdc start
```

A *KDC* adatbázis rendszeres elterjesztése érdekében indítsunk egy *cron* parancsot a *kdc.pelda.com*-on. *Jason Garman*nek (és az *O'Reilly* által megjelentetett *„Kerberos: The Definitive Guide"* című könyvének) köszönhetően kezünkben van egy működő *cron* parancs.
Kézenfekvő, hogy ezt a szkriptet óránként indítsuk a */etc/cron.hourly* könyvtárból. Ezek után *Kerberos* adatbázisunk biztonságosan vezetődik át a főfiókból a szolgagépek sokaságára. Ha a főfiók elromlik, lehetőségünk van arra, hogy valamelyik szolgagép könnyen-gyorsan átvegye a feladatát, minimális adatvesztéssel (vagy szerencsés esetben anélkül).
Ha már át tudjuk vezetni a *Kerberos* változásokat egy-egy szolgagépre, akkor nyilvántartásba vehetjük őket a *krb5.conf* fájlban, mint érvényes *KDC*-ket.

Az OpenLDAP biztonságos átvezetése

Minden fontos rendszerben kerülnünk kell az egy pontból eredő hibaforrásokat, *SPOF*-okat (*Single Point of Failure; egypontos hiba*). Problematikus lenne csak egyetlen helyen tárolni az *LDAP* címtárunkat; nem kevés kritikus információ veszne el hiba esetén, sőt a felhasználóink még be sem tudnának jelentkezni, lehetetlenné válna az e-mailek megnézése és számos egyéb napi teendő. Az *LDAP* címtár átvezetése ezt küszöböli ki.
Replikáljuk tehát az *LDAP* címtárat az *ldap.pelda.com*-ról a *kdc.pelda.com*-ra. Az OpenLDAP-nak van is egy háttérprogramja (démonja), ami pontosan ezért felel: a *slurpd*. Sajnos a *slurpd*-nek nincs olyan beállítási lehetősége, amivel meg lehetne neki adni, melyik *Kerberos keytab* fájlt kellene használnia, így szükség lesz egy kis kézimunkára. Szerkesszünk bele a *slapd.conf* fájlba a *ldap.pelda.com*-on, megadva a *replogfile* és *replica* opciókat, majd indítsuk újra a *slapd*-t.
Létre kell hoznunk egy *Kerberos* alapú *LDAP* főszolgáltatást, egy *SSL* tanúsítványt és egy kulcsot a *kdc.pelda.com* számára, ahogyan azt a *ldap.pelda.com* esetében is tettük, és a *slapd.conf* fájlt is be kell állítanunk ugyanitt. Ez szinte ugyanolyan, mint amilyet az *ldap.pelda.com*-on készítettünk, néhány kulcsfontosságú különbséggel.
Ugyanabból az okból, mint ami miatt csak egyetlen *Kerberos* főkiszolgálónk van, itt is csak egyetlen *LDAP* címtárat tartunk naprakészen, és ezen hajtjuk végre a változtatásokat. Az egyetlen felhasználó, akinek lesz jogosultsága írni a szolgagépek címtárába, az alábbi módon írható le:

```
uid=host/ldap.pelda.com,cn=GSSA
➥PI,cn=auth
```

Ő nem más, mint a főfiók *Kerberos* gazdája; így a szolgagépek hozzáférés-szabályozó listáit (*ACL*-jeit) jóval szigorúbbra kell szabni. A *slapd*-nek arról is kell tudnia, hogy az *updatedn* és *updateref* opciók által megadott módon *ki* fog a *slurp* révén frissítéseket küldeni.
Most újra irányítsuk figyelmünket az *ldap.pelda.com*-ra. Létre kell hoznunk a */etc/conf.d/slurpd* fájlt, vagy be kell állítanunk a `KRB5CCNAME` változót, mielőtt a megfelelő szkript elindítja a *slurpd*-t.
Ezek után beszerezzük az indításhoz szükséges *Kerberos* igazolványokat (*credentials*):

```
# KRB5CCNAME=/var/
# run/slurpd.krb5cache
# /usr/bin/kinit -k
```

Majd az egész címtárat kiíratjuk egy fájlba:

```
ldap# /etc/init.d/slapd stop
ldap# /usr/sbin/slapcat -l
➥/tmp/slavedump.ldif
ldap# /etc/init.d/slurpd start
```

Minthogy a *slurpd* csak a főfiókra hat, nekünk kell benépesítenünk a szolgagépek címtárait a főfiók tartalma alapján. Ezt úgy tesszük meg, hogy a */tmp/slavedump.ldif* fájlba kiírt főfiók-adatbázist (amit az imént már elkészítettünk) átmásoljuk a *kdc.pelda.com*-ra, ahol a fájl beolvasása után elindíthatjuk a *slapd*-t:

```
kdc# /usr/sbin/slapadd -l
➥slavedump.ldif
kdc# /etc/init.d/slapd start
ldap# /etc/init.d/slapd start
```

Ellenőrizzük, hogy a szolgagép címtára megfelelő-e:

```
# ldapsearch -H
# ldap://kdc.pelda.com -ZZ
```

Próbáljuk ki, hogy jól működik-e az átvezetés. Módosítsunk vagy adjuk hozzá új adatot az *ldap.pelda.com* címtárához, majd keressünk rá a *kdc.pelda.com*-on, hogy megbizonyosodjunk arról, hogy a változtatások átvezetődtek-e.
Ha meggyőződtünk arról, hogy a *slurpd* működik, hozzunk létre egy

alkalmas ***cron*** parancsot az ***ldap.pelda.com***-on, hogy meggátoljuk az igazolványok elévülését. Az igazolványok érvényességének alapértelmezett ideje tíz óra, így ha pl. nyolc óránként futtatjuk a ***cron*** parancsot, az megfelelő lesz. Végül fel kell vennünk a ***kdc.pelda.com***-ot az érvényes *LDAP* kiszolgálók közé az ***nss_ldap*** számára. Azaz: be kell illesztenünk a ***kdc.pelda.com***-ot abba a kiszolgálófelsorolásba, ami a */etc/ldap.conf „host”* („gazdagép”) opciójában szerepel.

A Postfix levélkezelő beállítása

Postfix levélkezelőt (***mail transport agent, MTA***) fogunk használni. A 2.1.5-ös verziójú ***Postfixben*** már jól kiépített támogatás található az SASL hitelesítésre, valamint olyan *LDAP* finomságok támogatására, mint az álnevek (***aliasok***). Mivel a ***Postfix*** beállításának alapoktól történő bemutatása túlmutatna e cikk keretein, most csak azzal foglalkozunk, hogy hogyan lehet rávenni a programot az ***SASL*** és a ***TLS*** használatára. A ***Postfix*** részletes beállítására vonatkozóan: információk a cikkhez tartozó források között.
A ***Postfixnek*** két fő konfigurációs fájlja van: a */etc/postfix/main.cf* és a */etc/postfix/master.cf*. A ***main.cf*** elsősorban a bejövő levelek fogadásáért felelős, míg a ***master.cf*** inkább a levélkézbesítő programok (***mail delivery agent, MDA***) működtetéséért.
Egy példa ***main.cf*** megtekinthető a cikkhez tartozó források között, de a részletek megértéséhez érdemes ismerni a ***Postfix*** dokumentációját és weboldalát.
Három fő kulcsszó határozza meg azt, hogy *SMTP* (***Simple Mail Transfer Protocol – kommunikációs protokoll az e-mailek továbbítására***) kiszolgálónk hogyan értekezzen más *SMTP* kiszolgálókkal: ***smtp_sasl_auth_enable, smtp_use_tls*** és ***smtp_tls_note_starttls***. Ha *SMTP* kiszolgálónk ki lesz téve az internet viharainak, akkor ezeket a változókat a lehető legrugalmasabb módon kell beállítani, hogy biztosan sikerüljön más *SMTP* kiszolgálókkal a kapcsolatfelvétel. Ha ez csak egy belső *SMTP* kiszolgáló, akkor viszont biztonságosabbra lehet szabni ezeket a beállításokat.
Az érdekesebb feladat annak beállítása, hogy miként adjuk meg a felhasználóink és számítógépeink kapcsolódását a levélkezelőnkhöz a levelek elküldésekor. Néhány egyéb opció, amit ezzel kapcsolatban jó ismerni: ***smtpd_sasl_auth_enable, smtpd_sasl_security_options, smtpd_sasl_tls_security_options, smtpd_use_tls, smtpd_tls_cert_file, smtpd_tls_key_file*** és ***smtpd_tls_auth_only***.
Ha *IMAP* rendszerű levélkézbesítést használunk, akkor győződjünk meg arról, hogy be van-e állítva a ***master.cf***-ben a ***mailbox_transport*** változó értéke, valamint az ***smtp*** és ***cyrus*** átviteli (***transport***) mechanizmus.
Az *OpenLDAP*-hez hasonlóan a ***Postfix*** is kerberizált; *SASL*-t használ a hitelesítési képesség-egyeztetéskor és *SSL* segítségével tudja biztosítani az adatátvitelt. A ***Postfix*** biztonságossá tételéhez, az *SASL* használatára való beállításához lesz néhány teendőnk a ***main.cf*** módosításán túl is. ***Először*** létrehozunk egy *SSL* tanúsítvány/kulcs párt és elhelyezzük e két összetevőt a */etc/ssl/postfix/smtp-cert.pem* és a */etc/ssl/postfix/smtp-key.pem* fájlba, miközben megbizonyosodunk arról, hogy a ***postfix*** felhasználó és a ***mail*** csoport tulajdonában vannak, és hogy a kulcs csak a ***postfix*** felhasználó számára olvasható.
Ezután elkészítünk egy főfiókot a ***mail.pelda.com*** számára, és elmentjük a normál helyére. Egy főszolgáltatást is létrehozunk, ***„smtp/mail.pelda.com@CI.PELDA.COM”***-ként, ezt pedig a */etc/postfix/smtp.keytab*-ba mentjük el. Ezt a fájlt a ***root*** felhasználó tulajdonában kell tartani, és ugyanolyan jogosultságokkal felruházni, mint az ***smtp-key.pem*** fájlt. Ezek után még létre kell hoznunk egy *SASL* konfigurációs fájlt */etc/sasl2/smtpd.conf* néven, és át kell szerkesztenünk a */etc/conf.d/saslauthd*-t. A ***Postfix*** a ***saslauthd*** háttérprogramot használja arra, hogy információt kapjon a hitelesítési mechanizmusokról. A fenti két fájl adja az *SASL* tudtára, hogy hogyan ellenőrizze a jelszavakat, milyen mechanizmusok támogatottak, és mi legyen a minimálisan használt biztonsági szint. A ***minimum_layer*** felvehető értéke megegyezik az *OpenLDAP*-ben megadható biztonságossági faktoréval (***„Security Strength Factor”, SSF***).
Végül pedig a */etc/conf.d/postfix* fájllal megmondjuk a ***Postfixnek***, merre találja a ***Kerberos keytab*** fájlját. (Vagy, mint ahogy eddig már többször láthattuk: a ***Postfix*** indítása előtt a megfelelő indító szkriptben létrehozott `KRB5_KTNAME` változó segítségével is megtehetjük ugyanezt). Ha mindezekkel végeztünk, elindíthatjuk a ***saslauthd***-t és a ***Postfixet*** indító szkripteket.
Az *LDAP* nem pusztán a személyazonosság-szervezés és hitelesítés miatt hasznos, hanem amiatt is, mert a ***Postfix*** számára átadható ***álnevek*** (***aliases***) szótárát is tudja kezelni. Egyszerű használni és karbantartani, és feleslegessé teszi azt, hogy minden változáskor újrageneráljuk az álnévadatbázist. Címtárunk akkor válik először igazán erős eszközzé, amikor az álnévszótárat is rendelkezésére bocsátjuk. Ezt azzal tehetjük meg, hogy átadjuk a ***misc.schema***-t a ***slapd*** konfigurációnak, majd létrehozunk a címtárban egy elágazást (***branch***) az álnevek számára. Használjuk ezt:

```
ou=aliases,o=ci,dc=pelda,dc=com
```

(Itt `ou` = ***organizational unit***; a többi rövidítést lásd fentebb). Az utolsó feladatrész abból áll, hogy megmondjuk a ***Postfixnek***, hogy az *LDAP*-ot használja az álnevek beazonosításához. Ezt az ***ldap:/etc/postfix/aliases.cf***-nek a ***main.cf***-beli ***alias_maps*** opciójához való beírásával tehetjük meg, valamint ezzel párhuzamosan a */etc/postfix/aliases.cf* fájl létrehozásával, ami megadja, hogy hogyan kell az *LDAP*-hez kapcsolódni, s hogy hol is vannak az álnevek az *LDAP*-ben. Újraindítjuk a ***slapd***-t, majd a ***Postfix***-et; íme, készen állunk arra, hogy létrehozzunk egy e-mail álnevet. Hozzunk létre egy *LDIF* fájlt, nevezetesen az ***alias.ldif***-et, és vegyük fel a címtárba. Tá-dáááám! Készen vagyunk!

A cyrus IMAP levélkézbesítő beállítása

A ***cyrus IMAP*** levélkézbesítő program 2.2.10-es verzióját fogjuk használni. (***A cyrus egy jól skálázható vállalati levelezőrendszer, mely megbirkózik sokféle szabványra épülő technológiával; használható***

egyedülálló gépeknél, de akár óriási, centralizált intézeti hálózatban is – a ford.) A *cyrus IMAP* kiszolgáló részletes beállításának bemutatása túlmutatna e cikk keretein, de működőképes példák fellelhetők a források között. A *cyrus IMAP* kiszolgálót ugyanaz a csoport fejlesztette, aki a *cyrus SASL*-t is, így az *SASL* és az egyszeri bejelentkezés az elvárásoknak megfelelően működik.

A *Postfixhez* hasonlóan a *cyrus IMAP*-nak is két konfigurációs fájlja van: a */etc/imapd.conf* és a */etc/cyrus.conf*. Most csak a */etc/imapd.conf*-fal foglalkozunk. Itt is adott néhány előfeltétel: *SSL* tanúsítvány/kulcs pár, főfiók és főszolgáltatás; ez utóbbit hívjuk így: *„imap/mail.pelda.com@CI.UCHICAGO.EDU”*, és tároljuk el a */etc/imap.keytab* fájlban. Az *SSL* beélesítéséhez adjuk meg alkalmas módon a *tls_ca_path, tls_cert_file* és *tls_key_file* opciókat. Az *SASL* használatához meg kell adnunk a *sasl_pwcheck_method, sasl_mech_list* és *sasl_minimum_layer* opciókat is. Ezek értékei egyezzenek meg azzal, mint amiket a *Postfix* számára megadtunk a */etc/sasl2/smtpd.conf* fájlban.

A *Postfixhez* hasonlóan a *cyrus IMAP* számára is meg kell mondanunk, hogy hol van a *keytab* fájlja, mégpedig a */etc/conf.d/cyrus* fájl révén. (Vagy, mint ahogy ez már szinte a könyökünkön jön ki: az *IMAP* háttérprogram indítása előtt a megfelelő indító szkriptben létrehozott KRB5_KTNAME változó segítségével is megtehetjük ugyanezt). Ha mindezzel elkészültünk, győződjünk meg arról, hogy valóban fut-e a *saslauthd* program, és ha igen, akkor futtassuk az *IMAP* indító szkriptjét.

Vízre bocsátás

Meglehetősen nagyot kaszáltunk rövid idő alatt, de megérte a fáradozás: egy biztonságos és jól skálázható vállalati címtár lett az eredménye. Lábra állítottunk egy rendszert, amely akár néhány, egy helyre tömörült felhasználót/számítógépet, akár a világban szétszóródott tízezernyit is képes kiszolgálni. Következő cikkemben azzal fogunk megbirkózni, hogy munkánk gyümölcseként miként tudunk *Linux* és *Apple OS X* kliensgépeket is hálózatunkba kapcsolni.

Köszönetnyilvánítás

Munkámban segítséget nyújtottak: *Matematikai, Információtechnológiai és Számítástudományi tanszék (Office of Advanced Scientific Computing Research, Office of Science)*, az *Amerikai Energiaügyi Minisztérium* a *W-31-109-ENG-38* számú szerződés szerint. További támogatást kaptam a *Chicagoi Egyetem Számítástudományi Intézetétől* és a *Nemzeti Tudományos Alaptól*.

Linux Journal 2006., 140. szám

Ti Leggett (leggett@mcs.anl.gov) a Futures Laboratory of the Mathematics and Computer Science Division rendszergazdája az Argonne National Laboratory-ban; emellett a Chicagoi Egyetem Számítástudományi Intézetében is dolgozik.

KAPCSOLÓDÓ CÍMEK

➲ www.linuxjournal.com/article/8581