HRVATSKA AKADEMSKA I ISTRAŽIVAČKA MREŽA
CROATIAN ACADEMIC AND RESEARCH NETWORK

Implementacija centraliziranog sustava autentikacije u Linux okruženjima

CCERT-PUBDOC-2004-04-68

CARNet CERT u suradnji s LS&S

Sigurnosni problemi u računalnim programima i operativnim sustavima područje je na kojem CARNet CERT kontinuirano radi.
Rezultat toga rada ovaj je dokument koji je nastao suradnjom CARNet CERT-a i LS&S-a, a za koji se nadamo se da će Vam koristiti u poboljšanju sigurnosti Vašeg sustava.

CARNet CERT, www.cert.hr - nacionalno središte za **sigurnost računalnih mreža** i sustava.

LS&S, **www.lss.hr**- laboratorij za sustave i signale pri Zavodu za elektroničke sustave i obradbu informacija Fakulteta elektrotehnike i računarstva Sveučilišta u Zagrebu.

Sadržaj

1. Uvod

Koncept autentikacije korisnika na Linux operacijskim sustavima prilično je jednostavan i dobro poznat. U osnovnoj konfiguraciji podaci o svim korisničkim računima i grupama pohranjeni su u `/etc/passwd`, `/etc/shadow` i `/etc/group` datotekama sustava, koje se kontaktiraju prilikom svakog pokušaja prijave korisnika u sustav kako bi se provjerio njegov identitet. Navedeno korisničko ime i zaporka uspoređuju se sa pripadajućim parametrima u `/etc/passwd`, odnosno `/etc/shadow` datotekama i, ovisno o rezultatu usporedbe, korisniku se dozvoljava, odnosno odbija pristup sustavu. Iako je ovakav pristup prilično intuitivan i jednostavan, isti sadrži neka ograničenja koja mogu predstavljati problem u okruženjima s većim brojem računala i korisnika. Naime, kod većih sustava poželjno je implementirati centralizirani sustav autentikacije korisnika, koji će omogućiti autentikaciju na većem broju računala sa istim korisničkim imenom i zaporkom. Za razliku od Windows računala organiziranih u domene, gdje je sustav centralizirane autentikacije inicijalno omogućen, kod Linux sustava potrebno je nešto više truda kako bi se postigla slična funkcionalnost. Windows 2000 domene koje se baziraju na *Active Directory* servisu i Kerberos autentikaciji najbolji su primjer prednosti koje nudi centralizirani model autentikacije. Svi podaci o korisničkim računima, grupama, računalima i sl. pohranjeni su unutar centraliziranog imenika (baziranog na LDAP imeniku i X.500 protokolu), nad kojim se provode sve promjene i administracija. Osim što se ovakvim pristupom uvelike smanjuje vrijeme potrebno za održavanje sustava, korisnicima se omogućuje pristup različitim resursima informacijskog sustava sa istim autentikacijskim parametrima.

U ovom dokumentu biti će opisani postupci kojima je moguće implementirati sustav centralizirane autentikacije pomoću LDAP servisa u okruženjima baziranim na Linux operacijskom sustavu. Opisan je postupak podešavanja LDAP poslužitelja kao centralnog repozitorija podataka i autentikacijskog poslužitelja, kao i koraci koje je potrebno poduzeti na strani klijenta kako bi se isti poslužitelj koristio u svrhu autentikacije. Osim implementacije i konfiguracije samog sustava, također su opisani i osnovni pojmovi vezani uz LDAP i X.500 protokole, koncept PAM modula i *Name Service Switch* servisa te drugi elementi usko vezani uz ovo područje.

2. Uvod u LDAP

LDAP (eng. ***L**ightweight **D**irectory **A**ccess **P**rotocol*) protokol nastao je kao simplificirana inačica DAP (eng. ***D**irectory **A**ccess **P**rotocol*) protokola za pristup imeničkim servisima baziranim na X.500 specifikaciji.

X.500 standard opisuje osnovnu strukturu i način korištenja distribuiranih imeničkih servisa, a prva inačica istoga razvijena je 1988. godine od strane *International Telecommunications Union* (ITU) organizacije. Spomenuti standard je u to vrijeme bio razvijen s idejom da se omogući kreiranje jedinstvenog elektroničkog direktorija, jednostavnog za korištenje, koji bi svim korisnicima na Internetu omogućio pristup podacima pohranjenim u imeniku.

Velik broj danas vrlo popularnih mrežnih servisa bazira se upravo na spomenutoj specifikaciji (kao što je npr. *Active Directory* servis kod Windows 2000 operacijskih sustava), što je jedan od pokazatelja iznimno velikih mogućnosti sustava baziranih na ovom principu. Iako prednosti i kvalitete imeničkih servisa ponajviše dolaze do izražaja u velikim računalnim okruženjima gdje je potrebno voditi evidenciju i upravljati velikim brojem informacijskih resursa (računala, korisnici, korisničke grupe i sl.), isti se na identičan način može primijeniti i u manjim okruženjima. Neke od osnovnih karakteristika imeničkih servisa navedene su u nastavku:

- sustav optimiziran za čitanje podataka,
- distribuirana pohrana podataka,
- moguće proširivanje modela podataka koji se pohranjuju u imeniku,
- napredne mogućnosti pretraživanja,
- mogućnost replikacije između više poslužitelja.

Imenički servisi vrlo se često izjednačavaju sa bazama podataka, iako postoje značajne razlike između ova dva pristupa. Osnovna razlika je ta što su imenički servisi optimizirani za akcije čitanja podataka, dok baze podataka podrazumijevaju podjednaku učestalost akcija pisanja i čitanja. Samim time prisutne su i brojne druge razlike između implementacija ovih dvaju servisa, koje ovdje neće biti detaljnije razmatrane.

Koncept Windows 2000 domena bazira se na *Active Directory* imeničkom servisu, a o prednostima njegovog korištenja ne treba puno govoriti. Kasnije će u dokumentu biti više riječi o tome kako se OpenLDAP programski paket, besplatna implementacija LDAP protokola, može iskoristiti za uspostavu imeničkih servisa u Unix/Linux mrežnim okruženjima.

2.1. X.500

Osnovna ideja koja se skriva iza X.500 standarda je stvaranje centralnog repozitorija podataka, imenika, u kojem se pohranjuju informacije o pojedinim informacijskim resursima (objektima). Podaci unutar imenika pohranjuju se hijerarhijski u obliku stabla, slično kao što je to slučaj kod DNS-a (engl. ***D**omain **N**ame **S**ystem*) ili Unix/Linux datotečnog sustava. Na vrhu stabla nalazi se korijen direktorija, ispod kojeg su podaci pohranjeni u obliku objekata sa odgovarajućim imenom i značenjem.

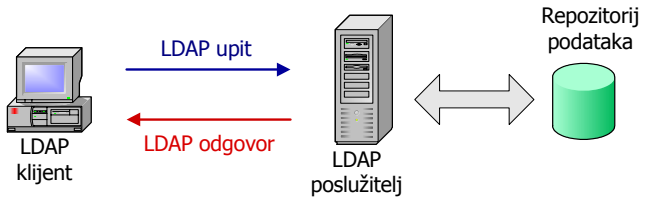
Cijela arhitektura bazira se na dobro poznatom klijent-poslužitelj modelu, u kojem su podaci distribuirani između jednog ili više poslužitelja, s kojima klijenti komuniciraju u obliku upita i odgovora. U X.500 terminologiji klijenti se nazivaju *Directory Service Agent* (u nastavku DUA), a poslužitelji *Directory System Agent* (u nastavku DSA) agenti. Između istih se komunikacija se odvija putem *Directory Access Protocol* (u nastavku DAP) protokola, dok poslužitelji međusobno komuniciraju putem *Directory System Protocol* (u nastavku DSP) protokola. DSA poslužitelj može se jednostavnije opisati kao centralizirana baza podataka u kojoj su informacije pohranjene prema točno definiranom modelu, opisanom X.500 standardom, i gdje se podaci prema potrebi mogu replicirati na druge DSA poslužitelje putem spomenutog DSP protokola. Distribuiranost sustava baziranog na X.500 modelu proizlazi upravo iz činjenice što se podaci imenika mogu nalaziti na nekoliko različitih, fizički odvojenih poslužitelja, između kojih se međusobno provodi sinkronizacija.

U ovom dokumentu neće se detaljnije ulaziti u opis X.500 standarda, budući da je isti prilično kompleksan, a nije ključan za razumijevanje sadržaja pokrivenog ovim dokumentom.

2.2. LDAP protokol

S obzirom na kompleksnost X.500 standarda i pripadajućeg DAP protokola za pristup imeničkim servisima baziranim na X.500 standardu, LDAP je razvijen kao "*lightweight*" inačica spomenutog protokola, od kuda i potječe njegov puni naziv. Kao što će biti pokazano u nastavku, koncepti su većim dijelom preuzeti iz X.500 standarda, iako su neki elementi pojednostavljeni kako bi se omogućila šira primjena i jednostavnija implementacija sustava baziranih na LDAP servisu. Uistinu, LDAP protokol prilično je jednostavan, kao i implementacija, odnosno uspostava servisa koji koriste njegove mogućnosti.

Klijent-poslužitelj model ukratko opisan u prethodnom poglavlju o X.500 standardu prikazan je ovdje na konkretnom primjeru LDAP servisa.

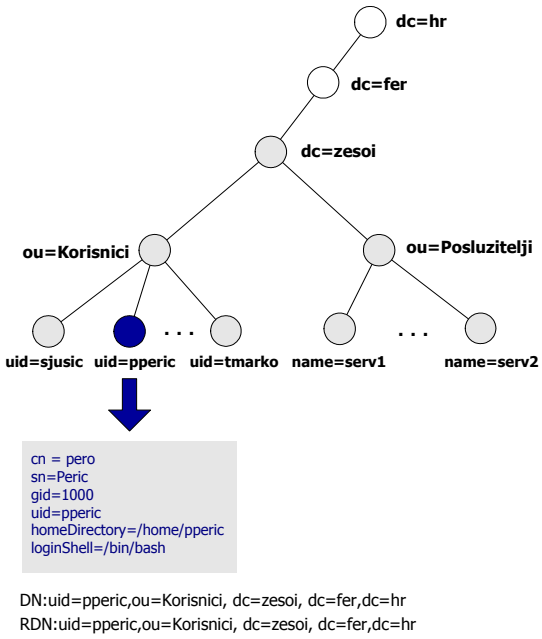


Slika 1: LDAP klijent-poslužitelj modul

LDAP klijenti prosljeđuju upite autoritativnom poslužitelju, koji ili vraća odgovor, ukoliko može doći do zatraženih podataka (slika a), ili klijentu vraća pokazivač (engl. *referral*) na drugi LDAP poslužitelj koji može razriješiti upit klijenta. Distribuiranost podataka između više poslužitelja jedna je od kvaliteta LDAP servisa, budući da se na taj način postiže redundancija komponenti sustava. U slučaju ispada jednog od poslužitelja, na raspolaganju je drugi koji u međuvremenu može opsluživati zahtjeve klijenata.

Model pohrane podataka kod LDAP servisa preuzet je u potpunosti od X.500 standarda. Podaci se pohranjuju u obliku objekata organiziranih u stablo (eng. *Directory Information Tree*, DIT), a svaki od objekata sastoji se od jednog ili više atributa kojim se detaljnije opisuju njegova svojstva (npr. objekt **Automobil** može sadržavati atribute **marka=Ford**, **boja=crvena**, **godina=2000** itd...). Unutar LDAP imenika svaki je objekt jedinstveno određen svojim DN (eng. ***Distinguished Name***) imenom kojim se opisuje njegova pozicija u strukturi imenika (nešto slično kao apsolutni put kod datotečnog sustava). DN ime tvori se od RDN (eng. ***Relative Distinguished Name***) imena objekta i puta do korijena imenika (prema analogiji s datotečnim sustavom ovo bi značilo da se apsolutni put datoteke tvori od njenog imena i puta do korijenskog `root` direktorija). Više o DN i RDN imenima dano je u nastavku dokumenta.

Hijerarhijska struktura imenika tradicionalno se kreirala prema geografskim lokacijama i organizacijskim cjelinama organizacije, dok se danas mnogo češće koristi struktura koja odgovara Internet domenama koje organizacija koristi. Osnovni razlog je taj što se time omogućuje integracija postojećeg DNS sustava i LDAP servisa što olakšava njegovu primjenu i općenito integraciju u postojeći informacijski sustav. Na slici je prikazan primjer LDAP imenika organiziranog prema konceptu Internet DNS domena.



Slika 2: Struktura LDAP direktorija

Organizacija stabla, tip podataka koji se pohranjuju, tip i značaj atributa za svaku klasu objekta i sl., parametri su koji se definiraju shemom LDAP imenika, koju je uvijek moguće prilagoditi okruženju u kojem se sustav implementira. Na taj način moguće je u LDAP imeniku centralizirano pohranjivati proizvoljne podatke s pripadajućim atributima, što predstavlja izvrsno rješenje u okruženjima gdje je većem broju klijentskih računala potrebno osigurati pristup strukturiranim podacima. Sljedećim primjerom detaljnije je opisan koncept objekata i atributa primijenjen kod LDAP servisa. U nastavku je dan primjer LDAP objekta s pripadajućim atributima.

```
dn: mail=sjusic@LSS.hr, dc=lss, dc=hr
objectclass: inetOrgPerson
cn: Sasa
sn: Jusic
uid: sjusic
mail: sjusic@lss.hr
telephoneNumber: 1 6129 956
```

Priloženi objekt jedinstveno opisuje korisnika `sjusic`. Objekt sadrži nekoliko različitih atributa koji ga pobliže opisuju, a cijeli zapis je jedinstveno definiran njegovim DN imenom, koje u ovom slučaju glasi `mail=sjusic@LSS.hr, dc=lss, dc=hr`. RDN ime objekta je u ovom slučaju njegova e-mail adresa (`mail=sjusic@LSS.hr`), iako je za RDN ime mogao biti odabran bilo koji drugi atribut objekta (npr. `uid=sjusic`). Koji će atribut biti odabran kao RDN ime objekta potpuno je ostavljeno na izbor dizajneru sustava, pod jedinim uvjetom da svaki objekt mora biti jedinstven unutar imenika. To znači da na istoj razini stabla ne smiju postojati dva ista objekta s istim RDN imenom.

Atributi koji smiju biti definirani za pojedini objekt nisu proizvoljni već se definiraju ***object class*** atributom. *Object class* atribut definira se kao dio svakog LDAP zapisa (instance objekta), čime se definiraju parametri obavezni za određeni objekt, kao i oni opcionalni. U gornjem primjeru, definirani objekt pripada klasi **inetOrgPerson** te kao takav mora zadovoljavati njena pravila, definirana shemom LDAP imenika. Svaki novi objekt kreiran u imeniku u stvari je instanca određene klase objekta, koja je definirana odgovarajućom shemom. Npr. svi objekti koji predstavljaju pojedine korisnike na sustavu instance su određene klase (npr. klasa **Korisnici**), kojom su definirani svi atributi koji se definiraju za pojedinog korisnika. Identičnu klasu objekata potrebno je definirati za svaki tip objekata koji se žele pohranjivati u imeniku.

Svaki tip objekta pripada jednoj ili više **object class** klasa čime se definira koji su atributi za taj objekt obavezni, a koji opcionalni. Definiranjem novih **object class** klasa struktura imenika može se jednostavno prilagoditi specifičnostima aplikacije. Shema LDAP imenika opisuje koji su atributi

obvezni, a koji opcionalni za pojedinu klasu objekata, kao i dozvoljene vrijednosti za svaki od definiranih atributa.

2.3. OpenLDAP

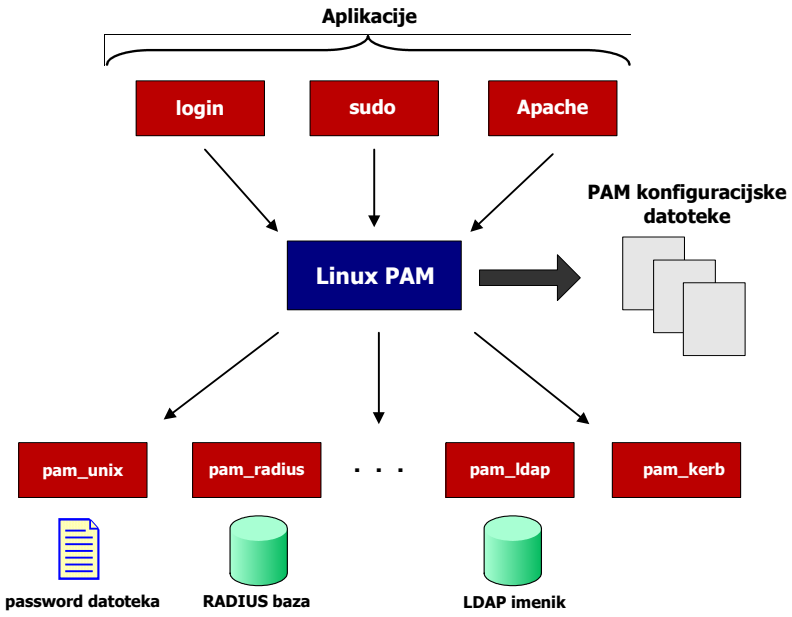
OpenLDAP (http://www.openldap.org/) je *open-source* implementacija LDAP protokola, koja sadrži sve elemente potrebne za uspostavu imeničkih servisa baziranih na LDAP protokolu. OpenLDAP podržava inačicu 3 LDAP protokola, različite mehanizme autentikacije, replikaciju podataka između poslužitelja, kao i podršku za različite tipove baza podataka za pohranu podataka (LDAP imenici za pohranu podataka koriste neku od podržanih tzv. *backend* baza podataka kao što su Berkeley DB, LDBM te ostale standardne Unix/linux baze podataka). Inačica 3 LDAP protokola u odnosu na inačicu 2 posjeduje sljedeće funkcionalnosti:

- autentikacija putem SASL biblioteke,
- podrška za SSL i TLS algoritme u svrhu zaštite integriteta i povjerljivosti podataka,
- podrška za Unicode kodiranje,
- preusmjeravanje klijenata na druge poslužitelje,
- podrška za različite *backend* baze i
- brojne druge napredne funkcionalnosti...

OpenLDAP programski paket implementira i klijentski (`ldapadd`, `ldapmodify`, `ldapsearch`) i poslužiteljski (`slapd`, `slurpd`) dio LDAP protokola, što znači da je isti paket dovoljan i na klijentskim i poslužiteljskim računalima.

3. PAM moduli

PAM (engl. ***P**luggable **A**uthentication **M**odules*) moduli predstavljaju skup biblioteka za autentikaciju korisnika na Linux/Unix operacijskim sustavima. Svaki od PAM modula (`pam_password`, `pam_kerb`, `pam_radius`, itd...) predstavlja određeni mehanizam autentikacije korisnika (datoteke sa zaporkama, Kerberos, RADIUS i sl.), a pojedine aplikacije, ovisno o namjeni i tipu, pozivanjem odgovarajućih modula iskorištavaju njihova svojstva. PAM moduli mogu se shvatiti kao sučelje između samih aplikacija i pojedinih mehanizama autentikacije podržanih na sustavu. Osim što se ovakvim pristupom dobiva iznimno modularna struktura, koja različitim programima omogućuje korištenje različitih mehanizama autentikacije, postupak prelaska na nove načine autentikacije vrlo je jednostavan i ne zahtjeva nikakve promjene na izvornom kod aplikacije. Dovoljno je promijeniti konfiguracijsku datoteku PAM programskog paketa u kojoj je za dotičnu aplikaciju potrebno navesti drugi modul. Bez koncepta PAM modula, svaka aplikacija ili servis morala bi sadržavati vlastiti kod za autentikaciju korisnika, što bi bilo vrlo neefikasno i nepraktično, pogotovo u slučaju potrebe prelaska na novi mehanizam autentikacije. Na sljedećoj slici (Slika 3) prikazan je koncept korištenja PAM modula:



Slika 3: Koncept PAM modula za autentikaciju

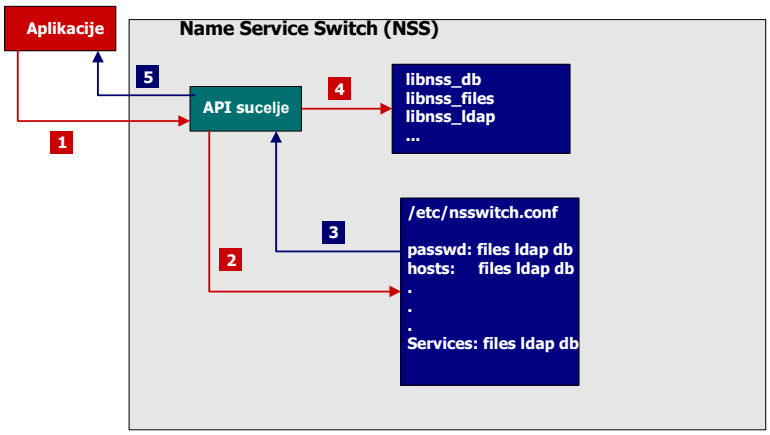
Korištenjem ovog koncepta moguće je bilo kojoj aplikaciji na sustavu (naravno, ukoliko ima podršku za PAM module) reći da umjesto `/etc/passwd` datoteke koristi LDAP imenik za autentikaciju korisnika. PAM modul za autentikaciju putem LDAP imenika naziva se `pam_ldap.so`, a detaljnije informacije o njemu je moguće pronaći na adresi http://www.padl.com/OSS/pam_ldap.html. Za implementaciju sustava autentikacije putem LDAP imenika na sustavu je potrebno obavezno instalirati `pam_ldap` modul.

U ovom dokumentu neće se ulaziti u detalje instalacije i konfiguracije PAM modula budući da to izlazi van područja razmatranja. Više informacija o tehnologiji PAM modula i pripadajućim funkcionalnostima moguće je naći na Web stranicama navedenim u poglavlju Reference (Poglavlje 7).

4. *Name Service Switch* (NSS)

Iako PAM moduli sustav autentikacije korisnika čine modularnim i vrlo praktičnim, isti ne omogućuju dolazak do svih informacija koje je potrebno pratiti o korisnicima za vrijeme njihovog rada na sustavu. Nakon uspješne prijave korisnika u sustav aplikacije zahtijevaju pristup različitim informacijama o njima (npr. login ljuska, *home* direktorij, UID i sl.), kako bi mogle uspješno izvršavati svoje zadatke. Iako su ove informacije tipično pohranjene unutar `/etc/passwd`, `/etc/shadow`, `/etc/group` i drugih sličnih datoteka, one mogu biti pohranjene i na bilo koji drugi način (baza podataka, LDAP i sl.).

Name Service Switch (NSS) sustav prihvaćen je kao API sučelje za jedinstven pristup podacima o korisnicima na sustavu, ali iz različitih izvora. Koji će se izvor koristiti ovisi o konfiguracijskoj datoteci NSS servisa (`/etc/nsswitch.conf`). Aplikacije putem pozivanja odgovarajućih funkcija dolaze do potrebnih podataka bez poznavanja detalja o lokaciji i načinu pohrane podataka. Na ovaj način moguće je pohranjivanje ključnih informacija sustava na različitim lokacijama, bez potrebe za izmjenama unutar samih aplikacija u slučaju bilo kakve promjene. Sve promjene definiraju se putem `/etc/nsswitch.conf` konfiguracijske datoteke. Na sljedećoj slici (Slika 4) prikazan je koncept primijenjen kod NSS servisa.

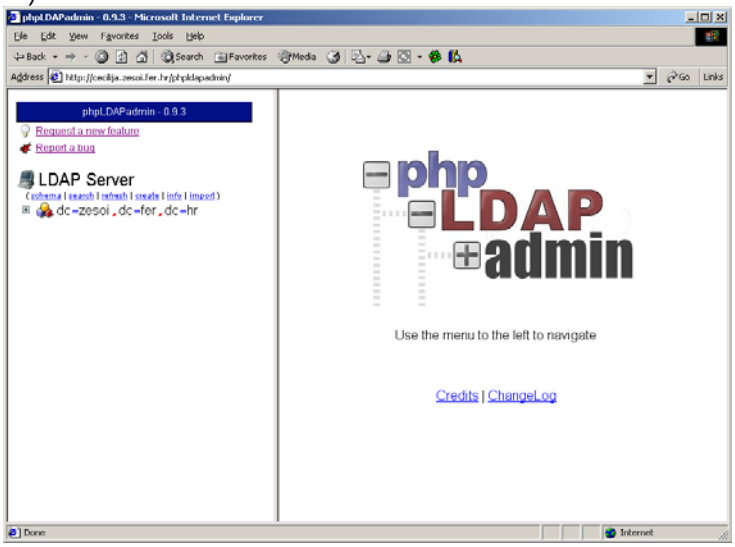


Slika 4: Koncept *Name Service Switch* servisa

Aplikacija putem definiranog API sučelja (korak 1) i postavki navedenih unutar `/etc/nsswitch.conf` konfiguracijske datoteke (korak 2) dolazi do potrebnih podataka pomoću odgovarajuće `libnss` biblioteke. Koja će se biblioteka koristiti za pristup pojedinim podacima ovisi o postavkama unutar spomenute datoteke, odnosno repozitoriju koji se koristi za pohranu podataka (za pristup podacima pohranjenim u LDAP imeniku koristi se modul `libnss_ldap`).
PAM moduli i NSS servis ovdje su spomenuti kao elementi neophodni za implementaciju sustava autentikacije na temelju LDAP servisa. Navedene PAM i NSS module aplikacije će koristiti za dolazak do podataka o korisnicima i drugim resursima pohranjenim unutar LDAP imenika.

5. Implementacija sustava

U nastavku dokumenta biti će opisani osnovni koraci koje je potrebno poduzeti kako bi se omogućila autentikacija korisnika putem LDAP servisa, odnosno OpenLDAP programskog paketa. Dokument će biti podijeljen na dva dijela, prvi u kojem su opisani postupci konfiguracije na strani poslužitelja i drugi s postupcima konfiguracije na strani klijenta.
Kako bi se omogućilo jednostavnije upravljanje i nadzor OpenLDAP poslužitelja, na testnom sustavu instalirana je phpLdapAdmin aplikacija (http://phpldapadmin.sourceforge.net/), koja omogućuje upravljanje LDAP imenikom putem Web sučelja. Na sljedećoj slici prikazano je sučelje spomenutog programa (Slika 5).



Slika 5: Glavno sučelje phpLdapAdmin programskog paketa

Spomenuta aplikacija predstavlja vrlo jednostavan i praktičan alat koji omoguće grafičko pregledavanje i upravljanje podacima unutar LDAP imenika. Osim što uvelike olakšava postupke unošenja novih i uređivanja postojećih podataka, aplikacija daje vrlo jasan i strukturirani pregled podataka u imeniku, što inače nije slučaj kod komandno linijskih alata kao što su `ldapmodify`, `ldapsearch`, `ldapadd` i sl.

5.1. Konfiguracija OpenLDAP poslužitelja

5.1.1. Uređivanje konfiguracijske datoteke

Nakon što je na sustavu uspješno instaliran OpenLDAP poslužitelj s odgovarajućim programskim paketima (`pam_ldap`, `nss_ldap`, `libldap2`,...), potrebno je podesiti odgovarajuće parametre unutar konfiguracijske datoteke OpenLDAP poslužitelja (`/etc/slapd.conf`). Neki od osnovnih parametara koje je potrebno podesiti navedeni su u nastavku:

```
database      ldbm
suffix        "dc=zesoi, dc=fer, dc=hr"
rootdn        "cn=root, dc=zesoi, dc=fer, dc=hr"
rootpw        {MD5}RBRJeG83acx3saMKMm7fLg==
directory     /var/lib/ldap
index         objectClass,uid,uidNumber,gidNumber     eq
index         cn,mail,surname,givenname               eq,subinitial
```

Opcijom `database` definira se tip baze u kojoj OpenLDAP programski paket pohranjuje podatke, `suffix` parametar predstavlja domenu organizacije, rootdn definira korisničko ime administratora LDAP poslužitelja sa odgovarajućom zaporkom u MD5 formatu (parametar `rootpw`). Parametar `directory` definira mjesto unutar datotečnog sustava u kojem se pohranjuju podaci imenika. Zaporku administratora u MD5 formatu moguće je dobiti zadavanjem sljedeće naredbe:

```
# slappasswd –h {MD5}
```

nakon čega će od korisnika biti zatraženo da unese odgovarajuću zaporku na temelju koje će biti generirana MD5 vrijednost. Osim MD5 algoritma moguće je koristiti i druge, npr. SHA-1, crypt, SSHA i sl.

Osim upravo opisanih parametara, `slapd.conf` konfiguracijska datoteka podržava i brojne druge parametre kojima je moguće preciznije kontrolirati način rada OpenLDAP poslužitelja, no one nisu toliko važne u ovom trenutku.

Nakon što su definirane osnovne postavke poslužitelja i nakon što je pokrenut odgovarajućom naredbom (`/etc/init.d/ldap start`), funkcionalnost poslužitelja moguće je testirati sljedećom naredbom:

```
# ldapsearch -x -b '' -s base '(objectclass=*)' namingContexts
version: 2

#
# filter: (objectclass=*)
# requesting: namingContexts
#

# search result
search: 2
result: 0 Success

# numResponses: 1
```

Ukoliko je odgovor sličan gore navedenome, OpenLDAP poslužitelj uspješno je instaliran i podešen. No, iako je u ovom trenutku poslužitelj u potpunosti funkcionalan, imenik još uvijek nije popunjen s odgovarajućim podacima. Populaciju imenika moguće je postići na nekoliko načina, a u nastavku dokumenta biti će opisan postupak kojim je moguće u imenik migrirati postojeće podatke sa sustava.

5.1.2. Kontrola pristupa

Jedan od vrlo važnih aspekata prilikom uspostave LDAP imenika je definiranje ovlasti pristupa imeniku. Budući da se unutar imenika vrlo često pohranjuju različiti podaci, različitih kategorija

prema povjerljivosti, potrebno je implementirati odgovarajuće sigurnosne mjere koje će onemogućiti neovlašteni pristup podacima u imeniku. Ovlasti pristupa imeniku moguće je postići izravnim dodavanjem odgovarajućih ACL listi u `slapd.conf` datoteku, iako je češća praksa da se unutar `slapd.conf` datoteke uključi zasebna datoteka u kojoj su navedena prava pristupa LDAP imeniku (`/etc/openldap/slapd.access.conf`). Uključivanje datoteke postiže se korištenjem ključne riječi `include`:

```
include         /etc/openldap/slapd.access.conf
```

Sintaksa definiranja ACL listi prilično je jednostavna i sastoji se od niza zapisa kojima se definira TKO ima pravo pristupa ČEMU. Postoji nekoliko ključnih riječi kojima je moguće definirati subjekte koji pristupaju podacima u imeniku. To su:

- `*` - svi korisnici,
- `users` - autenticirani korisnici,
- `self` – trenutni, prethodno autenticirani korisnik,
- `anonymous` – neautenticirani korisnici.

Na sličan način postoje i određene ključne riječi kojima se definira razina ovlasti prema pojedinim objektima pohranjenim u imeniku:

- `write` – ovlasti promjene atributa,
- `read` – ovlasti čitanja podataka u imeniku,
- `search` – ovlasti pretraživanja imenika,
- `compare` – ovlasti usporedbe atributa,
- `auth` – pristup zahtjeva autentikaciju klijenta,
- `none` - zabranjen pristup.

U nastavku je priložen sadržaj `slapd.access.conf` datoteke na kojem je detaljnije opisan spomenuti koncept kontrole pristupa podacima u imeniku.

```
# This is a good place to put slapd access-control directives
access to dn=".*,dc=zesoi,dc=fer,dc=hr" attr=userPassword
        by dn="cn=root,dc=zesoi,dc=fer,dc=hr" write
        by self write
        by * auth

access to dn=".*,dc=zesoi,dc=fer,dc=hr" attr=mail
        by dn="cn=root,dc=zesoi,dc=fer,dc=hr" write
        by self write
        by * read

access to dn=".*,ou=Korisnici,dc=zesoi,dc=fer,dc=hr"
        by * read

access to dn=".*,dc=zesoi,dc=fer,dc=hr"
        by self write
        by * read
```

Prvi zapis ograničava pristup `userPassword` atributu bilo kojeg zapisa unutar imenika. Administrator i vlasnik objekta (onaj koji se autenticira sa odgovarajućom zaporkom) ima pravo modifikacije atributa dok svi ostali pristupi zahtijevaju autentikaciju korisnika.

Drugi zapis dozvoljava modifikaciju adrese elektroničke pošte korisnika i njeno čitanje od strane svih drugih korisnika. Trećim zapisom navodi se da bilo koji objekt unutar organizacijske jedinice `Korisnici` mora biti dostupan samo za čitanje, dok posljednji zapis predstavlja tzv. "*catch all*" pravilo kojim se definiraju prava pristupa za sve ostale objekte imenika. Prilikom procesiranja upita pravila se analiziraju ozgora prema dolje i prvo pravilo koje zadovoljava upit uzima se kao važeće. Opisanim konceptom moguće je vrlo precizno definirati ovlasti pristupa pojedinim objektima unutar imenika, čime se može postići zadovoljavajuća razina sigurnosti na sustavu.

5.1.3. Migracija podataka

Nakon inicijalne uspostave OpenLDAP poslužitelja i osnovnih ACL lista potrebno je popuniti imenik odgovarajućim podacima. Iako je podatke moguće unositi ručno pomoću `ldapadd` naredbe, ovaj postupak mnogo je jednostavniji ukoliko se koristi `openldap-migration` programski paket.

Spomenuti paket sadrži niz skripti namijenjenih migraciji postojećih podataka sa sustava u LDAP imenik.
U tu svrhu potrebno je unutar `migrate_common.ph` datoteke definirati nekoliko osnovnih parametara koji će omogućiti prebacivanje podataka iz lokalnih datoteka sustava u LDAP imenik.

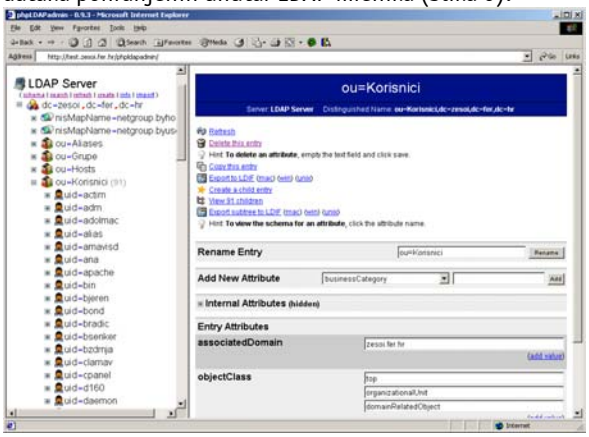
```
$DEFAULT_MAIL_DOMAIN = "zesoi.fer.hr";
$DEFAULT_BASE = "dc=zesoi,dc=fer,dc=hr";
$DEFAULT_MAIL_HOST = "cecilija.zesoi.fer.hr";
$EXTENDED_SCHEMA = 1;
```

Nakon podešavanje spomenutih parametara potrebno je odlučiti koji će se od postojećih podataka migrirati u LDAP imenik. Iako je najjednostavnije u ovom slučaju koristiti `migrate_all_online.sh` skriptu koja će u imenik prebaciti sve važnije podatke sa sustava (`/etc/passwd`, `/etc/shadow`, `/etc/group`, `/etc/services`, `/etc/protocols`, i sl.), preporučuje se provođenje detaljnijih analiza kako bi se odredili podaci koji se žele pohranjivati unutar LDAP imenika.
Podatke o servisima (`/etc/services`) i protokolima (etc/protocols) nema smisla pohranjivati u LDAP imeniku, budući da se radi o statičkim podacima kojima je jednostavnije pristupati izravno putem navedenih datoteka. Ovakva konfiguracija bi se i dodatno mogla negativno odraziti na performanse poslužitelja, obzirom na učestale upite klijenata.
No, podaci o korisničkim računima (`/etc/passwd` i `/etc/shadow`) svakako su podaci koje je potrebno pohranjivati u imeniku, pogotovo onda kada ga se želi iskoristiti za autentikaciju korisnika, kao što je to ovdje slučaj. No, detaljnijim razmatranjem opet se može zaključiti kako postoje određeni korisnički računi koje nema potrebe pohranjivat unutar imenika. Kao primjer mogu se navesti korisnički računi sustava kao što su `daemon`, `bin`, `adm` i sl., koji se nikada neće prijavljivati u sustav, kao i `root` korisnički račun za kojeg se smatra da ima lokalni značaj i da ga nema smisla dijeliti između svih računala (iako u nekim okolinama ovo može biti poželjno). Naravno, ovi detalji ovisit će o specifičnosti primjene i okruženju u kojem se sustav implementira.
S obzirom da se u ovom slučaju LDAP imenik koristi prvenstveno kao repozitorij podataka koji se koristi za autentikaciju korisnika, najveću važnost predstavljaju upravo podaci o korisničkim računima. U svrhu testiranja mogućnosti autentikacije putem OpenLDAP servisa, u imenik će biti uvezeni svi podaci sa sustava (korištenjem `migrate_all_online.sh` skripte). U nastavku je priložen rezultat izvršavanja spomenute skripte iz kojeg se može vidjeti postupak migracije podataka u LDAP imenik.

```
# /usr/share/openldap/migration/migrate_all_online.sh
Enter the X.500 naming context you wish to import into:
[dc=zesoi,dc=fer,dc=hr]
Enter the name of your LDAP server [ldap]: localhost
Enter the manager DN: [cn=manager,dc=zesoi,dc=fer,dc=hr]:
cn=root,dc=zesoi,dc=fer,dc=hr
Enter the credentials to bind with:
Do you wish to generate a DUAConfigProfile [yes|no]? No

Importing into dc=zesoi,dc=fer,dc=hr...

Creating naming context entries...
Migrating aliases...
Migrating groups...
Migrating hosts...
Migrating networks...
Migrating users...
Migrating protocols...
Migrating rpcs...
Migrating services...
Migrating netgroups...
Migrating netgroups (by user)...
Migrating netgroups (by host)...
adding new entry "dc=zesoi,dc=fer,dc=hr"

Importing into LDAP...
adding new entry "ou=Hosts,dc=zesoi,dc=fer,dc=hr"
adding new entry "ou=Rpc,dc=zesoi,dc=fer,dc=hr"
adding new entry "ou=Services,dc=zesoi,dc=fer,dc=hr"
adding new entry "nisMapName=netgroup.byuser,dc=zesoi,dc=fer,dc=hr"
adding new entry "ou=Mounts,dc=zesoi,dc=fer,dc=hr"
```

```
adding new entry "ou=Networks,dc=zesoi,dc=fer,dc=hr"
adding new entry "ou=Korisnici,dc=zesoi,dc=fer,dc=hr"
adding new entry "ou=Grupe,dc=zesoi,dc=fer,dc=hr"
adding new entry "ou=Netgroup,dc=zesoi,dc=fer,dc=hr"
adding new entry "ou=Protocols,dc=zesoi,dc=fer,dc=hr"
adding new entry "ou=Aliases,dc=zesoi,dc=fer,dc=hr"
adding new entry "nisMapName=netgroup.byhost,dc=zesoi,dc=fer,dc=hr"
adding new entry "cn=postmaster,ou=Aliases,dc=zesoi,dc=fer,dc=hr"
adding new entry "cn=MAILER-DAEMON,ou=Aliases,dc=zesoi,dc=fer,dc=hr"
adding new entry "cn=bin,ou=Aliases,dc=zesoi,dc=fer,dc=hr"
adding new entry "cn=daemon,ou=Aliases,dc=zesoi,dc=fer,dc=hr"
adding new entry "cn=games,ou=Aliases,dc=zesoi,dc=fer,dc=hr"
adding new entry "cn=ingres,ou=Aliases,dc=zesoi,dc=fer,dc=hr"
adding new entry "cn=nobody,ou=Aliases,dc=zesoi,dc=fer,dc=hr"
adding new entry "cn=system,ou=Aliases,dc=zesoi,dc=fer,dc=hr"
adding new entry "cn=nobody,ou=Aliases,dc=zesoi,dc=fer,dc=hr"
adding new entry "cn=system,ou=Aliases,dc=zesoi,dc=fer,dc=hr"
adding new entry "cn=toor,ou=Aliases,dc=zesoi,dc=fer,dc=hr"
adding new entry "cn=foo,ou=Aliases,dc=zesoi,dc=fer,dc=hr"
adding new entry "cn=falken,ou=Aliases,dc=zesoi,dc=fer,dc=hr"
adding new entry "cn=admin,ou=Aliases,dc=zesoi,dc=fer,dc=hr"
adding new entry "cn=manager,ou=Aliases,dc=zesoi,dc=fer,dc=hr"
adding new entry "cn=dumper,ou=Aliases,dc=zesoi,dc=fer,dc=hr"
adding new entry "cn=operator,ou=Aliases,dc=zesoi,dc=fer,dc=hr"
adding new entry "cn=leaf-2,ou=Protocols,dc=zesoi,dc=fer,dc=hr"
adding new entry "cn=rdp,ou=Protocols,dc=zesoi,dc=fer,dc=hr"
adding new entry "cn=irtp,ou=Protocols,dc=zesoi,dc=fer,dc=hr"
.
.
.
.
adding new entry "cn=iso-tp4,ou=Protocols,dc=zesoi,dc=fer,dc=hr"
adding new entry "cn=netblt,ou=Protocols,dc=zesoi,dc=fer,dc=hr"
adding new entry "cn=mfe-nsp,ou=Protocols,dc=zesoi,dc=fer,dc=hr"
adding new entry "cn=merit-inp,ou=Protocols,dc=zesoi,dc=fer,dc=hr"
adding new entry "cn=sep,ou=Protocols,dc=zesoi,dc=fer,dc=hr"
adding new entry "cn=3pc,ou=Protocols,dc=zesoi,dc=fer,dc=hr"
adding new entry "cn=idpr,ou=Protocols,dc=zesoi,dc=fer,dc=hr"
adding new entry "cn=xtp,ou=Protocols,dc=zesoi,dc=fer,dc=hr"
adding new entry "cn=ddp,ou=Protocols,dc=zesoi,dc=fer,dc=hr"
adding new entry "cn=idpr-cmtp,ou=Protocols,dc=zesoi,dc=fer,dc=hr"
adding new entry "cn=tp++,ou=Protocols,dc=zesoi,dc=fer,dc=hr"
ldap_add: Invalid DN syntax
        additional info: invalid DN

ldif_record() = 34
/usr/bin/ldapadd: returned non-zero exit status
```

Iz priloženog ispisa jasno se može vidjeti koji su podaci sa sustava prebačeni u LDAP imenik (ispis nije priložen u cjelovitosti zbog njegove duljine). Uz pomoć phpLDAPadmin moguće je vizualno analizirati dobivenu strukturu podataka pohranjenih unutar LDAP imenika (Slika 6).



Slika 6: Podaci učitani u LDAP imenik

Osim pregledavanja podataka, phpLDAPadmin aplikacija omogućuje i provođenje brojnih drugih akcija nad LDAP imenikom (dodavanje novih i modifikacija postojećih zapisa, odnosno atributa, izvoženje podataka (eng. *export*) i sl.), što može znatno olakšati administraciju i održavanje sustava, pogotovo za manje iskusne korisnike.
Iako se opisani postupak migracije čini vrlo jednostavnim, u konkretnim implementacijama isti će biti znatno složeniji s obzirom da će puno veću pažnju trebati posvetiti optimizaciji sustava i odabiru podataka koji se žele migrirati. U tu svrhu mogu poslužiti brojne skripte koje dolaze u sklopu `openldap-migration` programskog paketa, čija je osnovna namjena upravo migracija različitih tipova podataka u LDAP imenik. Priložene skripte omogućiti će prebacivanje određenog tipa podataka sa sustava u LDIF format, iz kojeg ih je moguće prebaciti u LDAP imenik korištenjem `ldapadd` naredbe. Za svaki tip podataka (korisnici, grupe, protokoli, računala i sl.) predviđena je zasebna skripta za migraciju, čime je vrlo precizno moguće kontrolirati cijeli postupak.

5.1.4. `Proxyuser` korisnik

Sljedeći korak u podešavanju LDAP poslužitelja za autentikaciju je dodavanje specijalnog `proxyuser` korisničkog računa, koji će imati dozvole čitanja nad `userPassworrd` atributima pojedinih korisnika. Zaseban korisnički račun sa ovlastima čitanja zaporki svih korisnika sustava potreban je kako bi se prilikom autentikacije klijenata mogao provjeriti njihov identitet. Kako će kasnije biti pokazano, `proxyuser` korisnički račun biti će potrebno definirati na strani klijenta kao onaj korisnički račun koji se upotrebljava za pristup podacima o zaporki klijenta.
`Proxyuser` zapis moguće je dodati na dva načina; korištenjem phpLDAPadmin aplikacije, što je jednostavniji način, ili korištenjem LDIF datoteke iz koje će se podaci pomoću `ldapadd` naredbe prebaciti u LDAP imenik.
Budući da je LDIF datoteka standardan postupak uvoženja podataka u LDAP imenik, ovdje će biti opisan na primjeru dodavanja novog zapisa koji opisuje `proxyuser` korisnički račun. U tu svrhu potrebno je kreirati sljedeću LDIF datoteku:

```
dn: cn=proxyuser, dc=zesoi,dc=fer,dc=hr
cn: proxyuser
sn: proxyuser
objectClass: top
objectClass: person
userPassword: {MD5}CY9rzUYh03PK3k6DJie09g==
```

iz koje je moguće kreirati novi zapis korištenjem `ldapadd` naredbe:

```
# ldapadd -x -D "cn=root,dc=zesoi,dc=fer,dc=hr" -W -f proxy.ldif
Enter LDAP Password:
adding new entry "cn=proxyuser, dc=zesoi,dc=fer,dc=hr"
```

Kako bi se `proxyuser` korisniku omogućio pristup `userPassword` atributima pojedinih korisnika potrebno je dodatno podesiti ACL liste kojima se definiraju ovlasti pristupa pojedinim objektima imenika (`/etc/openldap/slapd.access.conf`). Potrebno je definirati sljedeća pravila kojima se `proxy` korisniku omogućuje čitanje `userPassword` atributa.

```
access to dn=".*,dc=zesoi,dc=fer,dc=hr" attr=userPassword
        by dn="cn=root,dc=zesoi,dc=fer,dc=hr" write
        by dn="cn=proxyuser,dc=zesoi,dc=fer,dc=hr" read
        by self write
        by * auth
```

Nakon što su definirane ovlasti pristupa i nakon što je iznova pokrenut `slapd` poslužitelj, moguće je testirati funkcionalnost novo dodanog zapisa s pripadajućim ovlastima (korištenjem `ldapsearch` naredbe). Iz sigurnosnih razloga polje koje sadrži zaporku korisnika zamijenjeno je znakom #.

```
# ldapsearch -LL  -H ldap://localhost -b "dc=zesoi,dc=fer,dc=hr" -W
-x -D "cn=root,dc=zesoi,dc=fer,dc=hr" "(uid=sjusic)" userPassword
Enter LDAP Password:
version: 1

dn: uid=sjusic,ou=Korisnici,dc=zesoi,dc=fer,dc=hr
userPassword:: ##################################################=
```

5.2. Konfiguracija OpenLDAP klijenata

5.2.1. Uređivanje `/etc/ldap.conf` konfiguracijske datoteke

Nakon što su definirane osnovne postavke OpenLDAP poslužitelja i nakon što je obavljen postupak migracije podataka, potrebno je podesiti klijente koji će se autenticirati putem LDAP imenika (pritom se misli i na sam poslužitelj, budući da će isti također koristiti LDAP poslužitelj za autentikaciju; onaj lokalni).

Za razliku od podešavanja poslužitelja, gdje je potrebno uređivati `slapd.conf` konfiguracijsku datoteku, podešavanje klijenata zahtjeva uređivanje `ldap.conf` datoteke. U nastavku su navedeni neki od značajnijih parametara koje je potrebno podesiti kako bi se omogućila autentikacija pomoću udaljenog LDAP poslužitelja:

```
host 161.53.64.145
base dc=zesoi,dc=fer,dc=hr
rootbinddn cn=proxyuser,dc=zesoi,dc=fer,dc=hr
pam_filter objectclass=posixaccount
pam_login_attribute uid
pam_member_attribute gid
pam_template_login_attribute uid
pam_password md5
nss_base_passwd ou=Korisnici,dc=zesoi,dc=fer,dc=hr?one
nss_base_shadow ou=Korisnici,dc=zesoi,dc=fer,dc=hr?one
nss_base_group ou=Group,dc=zesoi,dc=fer,dc=hr?one
nss_base_hosts          ou=Hosts,dc=zesoi,dc=fer,dc=hr?one
nss_base_services       ou=Services,dc=zesoi,dc=fer,dc=hr?one
nss_base_networks       ou=Networks,dc=zesoi,dc=fer,dc=hr?one
nss_base_protocols      ou=Protocols,dc=zesoi,dc=fer,dc=hr?one
nss_base_rpc            ou=Rpc,dc=zesoi,dc=fer,dc=hr?one
nss_base_ethers         ou=Ethers,dc=zesoi,dc=fer,dc=hr?one
nss_base_netmasks       ou=Networks,dc=example,dc=com?ne
nss_base_bootparams     ou=Ethers,dc=zesoi,dc=fer,dc=hr?one
nss_base_aliases        ou=Aliases,dc=zesoi,dc=fer,dc=hr?one
nss_base_netgroup       ou=Netgroup,dc=zesoi,dc=fer,dc=hr?one
```

Navedenim parametrima definirana je adresa udaljenog OpenLDAP poslužitelja, korisnički račun pod čijim se ovlastima pristupa (`proxy` korisnički račun) te ostali parametri koji su potrebni za autentikaciju klijenta. Zaporku `proxy` korisničkog računa potrebno je pohraniti u datoteku `/etc/ldap.secret`, kojoj je, s obzirom na činjenicu da sadrži zaporku u čistom tekstualnom obliku, potrebno pridijeliti odgovarajuće ovlasti.

```
-rw-------    1 root     root            7 Apr 28 06:09 ldap.secret
```

Ovako definiranim ovlastima samo će root korisnik na lokalnom sustavu imati pravo pregledavati i mijenjati sadržaj ldap.secret datoteke, što znači da također ima pravo pregledavanja zaporki svih korisnika čiji su podaci pohranjeni u LDAP imeniku.

Kako je spomenuto u Poglavlju 5.1.4, `proxyuser` korisnik ima pravo čitanja `userPassword` atributa svih korisnika. Nedostatak ovakve konfiguracije je taj da korisnici sami neće moći mijenjati vlastite zaporke korištenjem `passwd` naredbe. Razlog tome je taj što `proxyuser` korisnik, pod čijim ovlastima klijentska računala pristupaju OpenLDAP poslužitelju, nema ovlasti promjene userPassword atributa, već samo njegovog čitanja. Ovaj problem moguće je vrlo jednostavno ukloniti, tako da se unutar `/etc/openldap/slapd.access.conf` datoteke korisniku `proxyuser` pridijele `write` ovlasti nad `userPassword` atributom.

```
access to dn=".*,dc=zesoi,dc=fer,dc=hr" attr=userPassword
        by dn="cn=root,dc=zesoi,dc=fer,dc=hr" write
        by dn="cn=proxyuser,dc=zesoi,dc=fer,dc=hr" write
        by self write
        by * auth
```

Osim prednosti koje ovakva konfiguracija nudi (mogućnost promjene zaporke korištenjem `passwd` naredbe), ista također povlači i određene sigurnosne probleme. Naime, u ovom slučaju `root` korisnik na lokalnom sustavu (koji ujedno i ne mora biti `root` korisnik na LDAP poslužitelju) ima ovlasti promjene zaporki svih korisničkih računa koji su pohranjeni u LDAP imeniku. Administratoru sustava

ostavlja se odluka da procijeni okruženje u kojem se sustav implementira te da u skladu s time donese potrebne odluke o načinu na koji će sustav biti implementiran.

5.2.2. Podešavanje NSS servisa

Nakon podešavanja `ldap.conf` konfiguracijske datoteke potrebno je podesiti NSS (*Name Service Switch*, Poglavlje 4) servis, kako bi se koristio LDAP poslužitelj kao repozitorij podataka, a ne lokalne datoteke, kao što je to slučaj u inicijalnoj konfiguraciji.
U nastavku je priložen primjer `/etc/nsswitch.conf` konfiguracijske datoteke u izvornom obliku:

```
passwd:     files nisplus nis
shadow:     files nisplus nis
group:      files nisplus nis
hosts:      files nisplus nis dns
```

Ovakvom konfiguracijom za dolazak do odgovarajućih podataka sustav koristi prvo lokalne datoteke (`/etc/passwd`, `/etc/shadow` i `/etc/group`), nakon toga `nisplus` servis, a nakon toga `nis`. Za razrješavanje IP adresa računala, kao dodatni izvor podataka naveden je DNS servis.
Ukoliko se za autentikaciju korisnika želi koristiti LDAP servis, potrebno je unijeti odgovarajuće izmjene unutar `/etc/nsswitch.conf` datoteke, na način kako je to opisano u nastavku:

```
passwd:     files ldap nisplus nis
shadow:     files ldap nisplus nis
group:      files ldap nisplus nis
```

Nakon opisanih promjena sustav će podatke tražiti prvo u lokalnim datotekama, a nakon toga u LDAP imeniku, koji je opisan parametrima unutar `/etc/ldap.conf` datoteke. Funkcionalnost novih postavki moguće je provjeriti korištenjem naredbe `getent`, na način kako je to pokazano u nastavku.

```
# getent hosts
# getent group
# getent passwd
# getent shadow
```

Svaka od navedenih naredbi ispisati će podatke koji su dostupni putem lokalnih datoteka, a nakon toga i putem udaljenog OpenLDAP poslužitelja. Iz sigurnosnih razloga ovi podaci nisu prikazani. Za podatke koji su pohranjeni i u lokalnim datotekama i udaljeno na LDAP poslužitelju, u ispisu će biti prikazana oba zapisa.
Iz modificirane inačice `nsswitch.conf` konfiguracijske datoteke može se vidjeti kako se podaci o imenima računala ne dohvaćaju sa LDAP poslužitelja, budući da je u tu svrhu uspostavljen DNS servis.

5.2.3. Podešavanje PAM sustava

Iako je za sam postupak autentikacije putem LDAP servisa dovoljno promijeniti konfiguraciju NSS servisa, kao što je to opisano u prethodnom poglavlju, za neke dodatne funkcionalnosti (kao što je promjena zaporke `passwd` naredbom) potrebno je uređivanje i PAM konfiguracijskih datoteka. Naravno, u tom slučaju potrebno je `proxyuser` korisničkom računu pridijeliti `write` ovlasti nad `userPasswod` atributom.
Podešavanje PAM sustava da koristi LDAP imenik vrlo je jednostavno. Na Linux Mandrake operacijskom sustavu, koji je korišten u svrhu testiranja postupaka autentikacije putem LDAP servisa, potrebno je učiniti odgovarajuće promjene unutar `system-auth` datoteke unutar `/etc/pam.d` direktorija.

```
#%PAM-1.0

auth        required      /lib/security/pam_env.so
auth        sufficient    /lib/security/pam_unix.so likeauth nullok
auth        sufficient    /lib/security/pam_ldap.so use_first_pass
auth        required      /lib/security/pam_deny.so

account     required      /lib/security/pam_unix.so
account     sufficient    /lib/security/pam_ldap.so

password    required      /lib/security/pam_cracklib.so retry=3
minlen=2  dcredit=0  ucredit=0
password    sufficient    /lib/security/pam_unix.so nullok
use_authtok md5 shadow
password    required      /lib/security/pam_deny.so
```

```
password    sufficient    /lib/security/pam_ldap.so use_authtok

session     required      /lib/security/pam_limits.so
session     required      /lib/security/pam_unix.so
session     optional      /lib/security/pam_ldap.so
```

Ukoliko se za želi omogućiti kreiranje odgovarajućeg *home* direktorija za svakog korisnika koji se prijavi u sustav, to je moguće postići korištenjem `pam_mkhomedir.so` PAM modula. U tom smislu potrebno je dodati sljedeću liniju u `system-auth` konfiguracijsku datoteku:

```
session     required      /lib/security/pam_mkhomedir.so \
skel=/etc/skel umask=022
```

Nakon dodavanja navedene linije, prilikom prijavljivanja korisnika koji nema svoj `home` direktorij na sustavu isti će biti kreiran nakon uspješno obavljene autentikacije. Primjer:

```
# su test
Creating directory '/home/test'.
Creating directory '/home/test/tmp'.

# id
uid=512(test) gid=513 groups=513

# ls -al
drwxr-xr-x    6 root     root         4096 Apr 28 08:17 ./
drwxr-xr-x   18 root     root         4096 Apr 29 01:17 ../
drwxr-xr-x    3 test    test          4096 Apr 28 08:23 test/
```

Rezultati izvršavanja `id` i `ls` naredbi pokazuju da je trenutno u sustav prijavljen korisnik test (UID 512), koji ne postoji na lokalnom sustavu. Također se može primijetiti da je i novo kreirani `/home/test` direktorij kreiran pod ovlastima tog korisnika, bez obzira što test korisnik ne postoji na lokalnom sustavu. Svi ovi parametri prikupljeni su od udaljenog OpenLDAP poslužitelja, putem kojeg se provodi autentikacija. Identičan `/home/test` direktorij biti će sa istim ovlastima kreiran i na svim drugim sustavima koji koriste isti OpenLDAP poslužitelj, što može biti vrlo praktično u okruženjima gdje jedan korisnik može koristiti više računala.

5.2.4. Autentikacija prema imenu računala

Prema upravo opisanom konceptu, svi korisnici čiji su parametri pohranjeni u LDAP imeniku mogu se prijaviti u bilo koji sustav koji za autentikaciju koristi isti LDAP poslužitelj. U određenim situacijama ovo može predstavljati sigurnosni problem, budući da se različita računala mogu koristiti za obavljanje različitih zadataka i da se ne želi dozvoliti da se na sva računala mogu prijaviti svi korisnici.
U tom slučaju potrebno je podesiti odgovarajuće parametre koji će omogućiti kontrolu nad time koji se korisnik može prijaviti u sustav sa kojeg računala. U tu svrhu potrebno je omogućiti `pam check host attr yes` parametar unutar `ldap.conf` datoteke.

```
pam_check_host_attr yes
```

Ovom direktivom `pam_ldap` modul će prilikom pokušaja prijave korisnika u sustav provjeriti vrijednost njegovog `host` atributa kako bi se utvrdilo da li mu je dozvoljena prijava s tog računala.
U svrhu dodavanja `host` atributa u LDAP imenik potrebno je kreirati odgovarajuću LDIF datoteku (`file-auth.ldif` u primjeru koji slijedi), na temelju koje će se korištenjem `ldapmodify` naredbe podaci učitati u LDAP imenik.

```
dn: uid=test,ou=Korisnici,dc=zesoi,dc=fer,dc=hr
changetype: modify
add: host
host: cecilija.zesoi.fer.hr

# ldapmodify -H ldap://localhost -D "cn=root,dc=zesoi,dc=fer,dc=hr"
-x -W -f file-auth.ldif
Enter LDAP Password:
modifying entry "uid=ihome,ou=Korisnici,dc=zesoi,dc=fer,dc=hr"
```

Dodavanjem novog zapisa korisniku test omogućuje se prijava u sustav samo sa lokalnog računala cecilija.zesoi.fer.hr na kojem je ujedno instaliran OpenLDAP poslužitelj. Pokušaj prijavljivanja korisnika test u sustav sa nekog drugog računala rezultirati će sljedećom porukom o grešci:

```
# su test
Access denied for this host
```

Na ovaj način moguće je vrlo precizno za sve korisničke račune definirati s kojih se računala mogu prijaviti u sustav, čime se dodatno može podići razina sigurnosti cijelog sustava. Naravno, ACL listama potrebno je ograničiti koji korisnici smiju modificirati koje atribute unutar LDAP imenika. Ukoliko bi se korisnicima omogućila promjena svih svojih atributa, cijeli koncept ne bi imao previše smisla, budući da bi korisnici sami mogli modificirati ograničenja nametnuta od strane administratora.

5.3. Korištenje SSL/TLS protokola

Razinu sigurnosti cijelog sustava autentikacije putem LDAP imenika moguće je dodatno podići korištenjem SSL, odnosno TLS protokola za enkripciju mrežnog prometa. Budući da se u postupku autentikacije mrežom šalju osjetljivi korisnički podaci, upotreba spomenutih protokola u ovom je slučaju preporučljiva. U tom smislu potrebno je kreirati odgovarajuće kriptografske ključeve i certifikate koje je nakon toga potrebno navesti u konfiguracijskoj datoteci poslužitelja (`slapd.conf`):

```
TLSCertificateFile      /etc/ssl/openldap/ldap.pem
TLSCertificateKeyFile   /etc/ssl/openldap/ldap.pem
TLSCACertificatePath    /etc/ssl/openldap/
TLSCACertificateFile    /etc/ssl/openldap/ldap.pem
```

Nakon toga potrebno je i na strani klijenta (`ldap.conf`) dodati sljedeće parametre:

```
ssl start_tls
```

Nakon toga potrebno je iznova pokrenuti OpenLDAP poslužitelj i sav promet između klijenta i poslužitelja biti će kriptiran TLS protokolom. Ukoliko ranije generirani certifikati iz određenog razloga nisu prikladni, moguće je generirati nove korištenjem `openssl` programskog paketa. Spomenuti postupak ukratko je opisan u nastavku:

```
# openssl genrsa -out ldap.key 1024
warning, not much extra random data, consider using the -rand option
Generating RSA private key, 1024 bit long modulus
.........................................++++++
....++++++
e is 65537 (0x10001)

# openssl req -new -key ldap.key -out ldap.csr
Using configuration from /usr/lib/ssl/openssl.cnf
You are about to be asked to enter information that will be
incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name
or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:HR
State or Province Name (full name) [Some-State]:Croatia
Locality Name (eg, city) []:Zagreb
Organization Name (eg, company) [Internet Widgits Pty Ltd]:LSS
Organizational Unit Name (eg, section) []:LSS Security
Common Name (eg, YOUR name) []:cecilija.zesoi.fer.hr
Email Address []:sasa.jusic@cecilija.zesoi.fer.hr

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:pasadoble
An optional company name []:FER
```

Prilikom generiranja zahtjeva za izdavanjem certifikata (`ldap.csr`, *Certificate Signing Request*) potrebno je unijeti odgovarajuće podatke o organizaciji za koju se certifikat izdaje (*Common Name*, *Country, Organization name* i sl.). Dobiveni zahtjev za izdavanjem certifikata prikazan je u nastavku:

```
-----BEGIN CERTIFICATE REQUEST-----
MIICFTCCAX4CAQAwgaYxCzAJBgNVBAYTAkhSMRAwDgYDVQQIEwdDcm9hdGlhMQ8w
DQYDVQQHEwZaYWdyZWIxDDAKBgNVBAoTA0xTUzEVMBMGA1UECxMMTFNTIFNlY3Vy
aXR5MR4wHAYDVQQDExVjZWNpbGlqYS56ZXNvaS5mZXIuaHIxLzAtBgkqhkiG9w0B
```

```
CQEWIHNhc2EuanVzaWNAY2VjaWxpamEuemVzb2kuZmVyLmhyMIGfMA0GCSqGSIb3
DQEBAQUAA4GNADCBiQKBgQCwssxlKvFExLe4L0IOE1hCT4+prmo+Jb8guofqBIxB
2y1bAJuON16RSdnWDGAzZ9R73qbo/+CocLwRl0d3AcCBMjT9caMKCdv6woRmDz1s
t6JTjig4Z8hi8lsL66+UovO6X20vG4gA3m35Df8nodl5OQQvjrTW5V6NGIjQN66S
NQIDAQABoC4wEgYJKoZIhvcNAQkCMQUTA0ZFUjAYBgkqhkiG9w0BCQcxCxMJcGFz
YWRvYmxlMA0GCSqGSIb3DQEBBAUAA4GBAFkCx+VXfVbRXKS+CdSWwF81eAP0n5mF
l5Y/abin8/PjwN2qph9bNISuZa/BNoS2ysCyFcywcP2tg3jXWQuAiRYL/CXcjhqr
a7dwwef8ZKoDHwjfU++/9MBxKw39jr8z9BxJlrpBUfJB7mhlv1As7yzuFptoWISo
6HtQYpwMToCk
-----END CERTIFICATE REQUEST-----
```

Na temelju generiranog zahtjeva potrebno je generirati valjani certifikat, potpisan od neke od CA (eng. *Certificate Authority*) organizacija (Thawte, Verisign i sl.), ili ga je moguć potpisati osobno, ukoliko primjena sustava to dozvoljava. Samopotpisivanje certifikata `openssl` programom prikazano je u nastavku.

```
# openssl x509 -req -in ldap.csr -out ldap.cert -CA ca.cert -CAkey
ca.key -CAcreateserial -days 365
Signature ok
subject=/C=HR/ST=Croatia/L=Zagreb/O=LSS/OU=LSS
Security/CN=cecilija.zesoi.fer.hr/Email=sasa.jusic@cecilija.zesoi.fe
r.hr
Getting CA Private Key
```

Rezultat izvođenja prikazane naredbe je datoteka `ldap.cert` koja sadrži certifikat potpisan privatnim ključem lokalnog CA-a kreiranog u ovu svrhu.

```
-----BEGIN CERTIFICATE-----
MIIDPjCCAiYCAQEwDQYJKoZIhvcNAQEEBQAwgaYxCzAJBgNVBAYTAkhSMRAwDgYD
VQQIEwdDcm9hdGlhMQ8wDQYDVQQHEwZaYWdyZWIxFTATBgNVBAoTDExTUyBTZWN1
cml0eTEMMAoGA1UECxMDTFNTMR4wHAYDVQQDExVjZWNpbGlqYS56ZXNvaS5mZXIu
aHIxLzAtBgkqhkiG9w0BCQEWIHNhc2EuanVzaWNAY2VjaWxpamEuemVzb2kuZmVy
LmhyMB4XDTA0MDQzMDE0NDc1MFoXDTA1MDQzMDE0NDc1MFowgaYxCzAJBgNVBAYT
AkhSMRAwDgYDVQQIEwdDcm9hdGlhMQ8wDQYDVQQHEwZaYWdyZWIxDDAKBgNVBAoT
A0xTUzEVMBMGA1UECxMMTFNTIFNlY3VyaXR5MR4wHAYDVQQDExVjZWNpbGlqYS56
ZXNvaS5mZXIuaHIxLzAtBgkqhkiG9w0BCQEWIHNhc2EuanVzaWNAY2VjaWxpamEu
emVzb2kuZmVyLmhyMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCwssxlKvFE
xLe4L0IOE1hCT4+prmo+Jb8guofqBIxB2y1bAJuON16RSdnWDGAzZ9R73qbo/+Co
cLwRl0d3AcCBMjT9caMKCdv6woRmDz1st6JTjig4Z8hi8lsL66+UovO6X20vG4gA
3m35Df8nodl5OQQvjrTW5V6NGIjQN66SNQIDAQABMA0GCSqGSIb3DQEBBAUAA4IB
AQD0QNLq08gZ2J5z8nuWMMksCqXolNR/Bu8nSScPfRiG0IdIjdYp2Y6jrihuyXMt
YATzJejdk8WY36I1xR6rD382KItSDOispZNjA2QL4fnZoguF7FLwREXPYhqDfNZh
YpEYVI3uDciWdn7bu7ZS5i3ioPIutj2JVnubVg0Dpyi/ws9dEHL0lkClEV57vJ4I
ka84z/2Pzscdltq7mlfW8RXZAQcudMobv64JS1VYWDGvzji8lKGpqSkxR5PjZh7M
H6QRNEwgwnLFwQMbOceCSF2ZmIi3H+gl8/HHm8I2980fzHs03vrZ8fmwkXgUPF6y
2LzbKhzRG8QHtjzF3vLPXOcd
-----END CERTIFICATE-----
```

Dobivene datoteke potrebno je uključiti u konfiguracijsku datoteku OpenLDAP poslužitelja na način kako je to ranije opisano.

6. Zaključak

Dokument opisuje postupak implementacije centraliziranog sustava autentikacije baziranog na LDAP servisu na Linux operacijskom sustavu. Opisani su osnovni koraci uspostave LDAP autentikacijskog poslužitelja, kao i koraci koje je potrebno poduzeti na strani klijenta kako bi se autentikacija provodila putem navedenog poslužitelja. Također su opisani osnovni koncepti, protokoli i servisi vezani uz ovo područje (X.500, LDAP protokol, PAM moduli, *Name Service Switch* i sl.), ne bi li se čitateljima na taj način olakšalo razumijevanje materije pokrivene ovim dokumentom.
Korištenjem OpenLDAP programskog paketa ispitane su osnovne funkcionalnosti i mogućnosti ovakvog sustava, popraćene konkretnim primjerima.

7. Reference

Mandrakesecure, Using OpenLDAP For Authentication, http://www.mandrakesecure.net/en/docs.php
O'Reilley, LDAP System Administration
LDAP Linux HOWTO, http://www.tldp.org/HOWTO/LDAP-HOWTO/