# 09. 기술문서 : LDAP을 이용한 계정통합, 각종 애플리케이션 연동

This page last changed on 2007년 07월 18 by joon.

LDAP

문태준

# LDAP 개략

- LDAP 용도는 무엇인가 : 읽기에 최적화되어있습니다. 디렉토리구조에 유용합니다. (인터넷회사에서 DHCP 로 ip할당하는 정보저장, 인증서 정보저장 등에 사용되고 있습니다)
- LDAP을 가지고 활용할 수 있는 것은?
- 사용자정보통합 : os계정, 이메일계정, ftp, http, outlook의 주소록등 통합가능. OS계정의 경우 호스트와 사용자 조합으로 접속제한을 할 수 있음.
- 참고로 윈도우즈의 Active Directory는 LDAP과 커버러스를 이용함. LDAP은 계정통합, 각종 정보통합에 사용을 하고 커버러스는 싱글사인온(SSO)에 사용을 함. 커버러스를 이용하여 네트워크를 통해 패스워스를 보내지 않고 키서버를 통하여 통신을 하고 티켓을 발급한 일정한 시간동안은 필요한 자원에 대한 별도 로그인이 필요없음.

# 문서소개

- 본 내용은 Redhat Enterprise Linux 3, CentOS4.4 에서 테스팅을 한 내용이며 다른 리눅스 배포판에서도 비슷하게 적용이 가능합니다. PAM 설정등은 시스템에 따라 다를 수 있습니다.
- LDAP 에 대한 소개가 아니므로 이에 대한 설명은 다른 문서를 참고하시기 바랍니다.
- openldap을 이용하여 계정통합을 하는 부분에 대한 자료는 여러가지가 있는데 이 문서는 거기에 추가로 필요한 상세한 내용을 담았습니다.
- LDAP을 이용한 사용자 인증 통합 (id, group, hosts)
- 사용자별, 호스트별 사용자 접속 제한
- 아이디, 그룹관리 프로그램(cpu)
- ldap replication (1 master, 1 slave)

- TLS 사용한 암호화통신
- nfs, autofs 이용한 사용자 홈디렉토리 공유
- outlook 등 주소록 활용
- 아파치 인증 활용
- 로그확인(syslog)
- gui 관리 프로그램
- NIS 기능으로 호스트 접근제한 추가

# 문서변경사항

- 2006.4.13 autofs 부분 스키마 추가부분
- 2006.4.12 autofs 부분 오타 수정. ACL 주의사항 추가
- 2006.3.30 SSL 관련부분 전체 정리 추가
- 2006.3.27 authconfig 프로그램 내용 보강
- 2007.3.26 아파치 웹인증부분 자료추가
- 2007.3.22 ssl/tls 연동시 클라이언트 설정 주의사항 추가 (CA인증서 복사)
- 2007.5.25 ldapseach 예제 추가

# 관련자료

## LDAP 초보자를 위한 기초자료

- LDAP에 대한 한글자료는 DSN의 자료 1개와 KLDP의 LDAP 하우투 및 기타 몇개의 문서가 있습니다. 상세한 내용은 영문자료를 보아야 합니다.
- http://database.sarang.net/?inc=read&aid=1243&criteria=ldap&subcrit=tutorials&id=&limit=20&keyword=&page=1 : LDAP의 모든것 ver 20011126. DSN에 2001년 올라왔던 ldap 전반적인 자료. 국내에 ldap 에 대한 한글자료가 별로 없는데 그나마 상세하게 ldap 에 대한 설명이 들은 한글문서입니다. 전체적으로는 LDAP기초부터 기본적인 사용법을 담고 있어 처음에 참고를 할 만 합니다.
- 위의 한글자료를 기초로 하여 약간 개정한 문서를 만들었습니다. 같은 URL을 참고하시면 됩니다. LDAP의 모든것2
- http://wiki.kldp.org/wiki.php/LinuxdocSgml/LDAP-HOWTO : KLDP LDAP 하우투자료
- O'REILLY 의 LDAP System Administration 서적 : LDAP 전반적인 설명을 담고 있으며 각종 애플리케이션을 ldap으로 통합하는 경우에 대한 상세한 자료를 제공하고 있음
- http://www.openldap.org/doc/admin23/ openldap 문서 : openldap에 대한 기본 사용법은 openldap 에서 제공하는 문서를 참고

## LDAP을 이용한 계정통합

- http://www.linuxjournal.com/article/8119 : OpenLDAP Everywhere - openldap 을 통한 계정통합 및 autofs 에 대한 내용. 상세한 세팅내용이 담겨있음. 주로 참고하였음.
- http://www-128.ibm.com/developerworks/library/l-openldap/index.html openldap 을 이용한 계정통합. 로그조정, relication, TLS 세팅에 대한 자료가 있음
- http://www.samag.com/documents/s=9494/sam0502a/0502a.htm : Centralized User Management with Kerberos and LDAP - kerberos, ldap을 이용한 사용자 통합에 대한 문서로 cpu 프로그램에 대한 소개가 있음
- http://www.linuxjournal.com/article/5505 : Highly Available LDAP - 공개 ha 프로그램을 이용하한 ldap ha 구성에 대한 내용임. ldap 구성에 대한 내용은 아니므로 도움은 되지 않을 듯 하지만 참고로 넣어두었음

## 기타 참고자료

- http://www.redhat.com/docs/manuals/dir-server/ 레드햇의 LDAP 문서. Administrator's Guide 등은 참고로 보면 좋을듯하며 Deployment Guide 는 ldap 설계에 대한 상세한 내용을 담고 있습니다.

- http://www.redhat.com/docs/manuals/dir-server/deploy/7.1/deployTOC.html Deployment Guide Red Hat Directory Server . 데이타 디자인, 스키마 디자인, 디렉토리 트리 디자인, 토폴리지 디자인, 리플리케이션 디자인, 보안 디자인, 튜닝 및 최적화, 운영관련 결정사항
- http://directory.fedora.redhat.com/ 페도리 디렉토리 서버. 레드햇에서 넷스케이프 디렉토리를 인수하여 제품화한 것이 레드햇 디렉토리 서버이며 이에 대한 공개버전이 페도리 디렉토리 서버입니다.

# 사전 확인사항

## 정책결정

LDAP 설계하기 : 데이타 디자인, 스키마 디자인, 디렉토리 트리 디자인, 토폴리지 디자인, 리플리케이션 디자인, 보안 디자인, 튜닝 및 최적화, 운영관련 결정사항
dc(suffix) 정하기 : 사용할 도메인
rootdn 의 패스워드 결정
계정정책 : UID, GID 범위

## 설치프로그램

RPM을 이용하여 설치
openldap-devel : openldap 과 연관된 프로그램을 개발할때 필요함. cpu 프로그램을 사용해야 할 경우 필요함
openldap : OpenLDAP 서버와 클라이언트 프로그램을 실행하기 위한 라이브러리
openldap-clients : client 프로그램
openldap-servers : server 프로그램
nss_ldap : NSS library and PAM module for LDAP

# ldap 서버설정

/etc/openldap/slapd.conf 에서 rootpw 를 추가함. 이를 통하여 root 권한 인증 사용함
아래 패스워드는 slappasswd 를 이용하여 생성함

```
[migration:root@localhost openldap]# grep -v "^#" slapd.conf
include         /etc/openldap/schema/core.schema
include         /etc/openldap/schema/cosine.schema
include         /etc/openldap/schema/inetorgperson.schema
include         /etc/openldap/schema/nis.schema
allow bind_v2

pidfile         /var/run/slapd.pid
argsfile        /var/run/slapd.args
loglevel        256

TLSCipherSuite HIGH:MEDIUM:+SSLv2
TLSCACertificateFile /etc/openldap/cacerts/cacert.pem
TLSCertificateFile /etc/openldap/slapdcert.pem
TLSCertificateKeyFile /etc/openldap/slapdkey.pem

database        bdb
suffix          "dc=samjung,dc=com"
rootdn          "cn=manager,dc=samjung,dc=com"
rootpw                {SSHA}aaaaaamoxk2Sswm8NbHZbCx9LxextJ

directory       /var/lib/ldap

index objectClass                       eq,pres
index ou,cn,mail,surname,givenname      eq,pres,sub
index uidNumber,gidNumber,loginShell    eq,pres
```

```
index uid,memberUid                     eq,pres,sub
index nisMapName,nisMapEntry            eq,pres,sub

cachesize       2000

access to dn.subtree="dc=samjung,dc=com" attr=userPassword
        by self write
        by* auth
access to dn.subtree="ou=people,dc=samjung,dc=com"
        by* read
access to dn.subtree="ou=group,dc=samjung,dc=com"
        by* read
access to dn.subtree="ou=hosts,dc=samjung,dc=com"
        by* read
access to*
        by* auth

replogfile /var/lib/ldap/openldap-master-replog
replica uri=ldap://cent.tunelinux.pe.kr:389
        suffix="dc=samjung,dc=com"
        binddn="cn=replica,dc=samjung,dc=com"
        credentials=xxxxxx
        bindmethod=simple
        tls=yes
```

위에서 suffix 를 조정하고 rootdn도 이와 맞추며 rootpw 를 설정하면 됨

```
# /etc/init.d/ldap start
Starting slapd:                                          [migration:  OK  ]
```

위에서 초기 세팅시 TLS 부분은 빼도 된다. ACI 는 사용자비밀번호는 자신만 바꿀수 있도록 하였고 people, group, hosts 정보는 누구나 읽을 수 있도록 하였다.
replication 부분도 초기 세팅시 빼도 된다.

database backend 모듈은 ldbm, bdb 등이 있다. bdb는 openldap 2.1부터 도입이 되었으며 Berkeley DB4 라이브러리만 사용하도록 맞추어져있다. bdb 가 ldbm에 비해 낫고 하는데 어떤 점이 나은지까지는 확인하지 않았다.

# 기본 정보 입력

## directory structure 생성

아래 내용을 top.ldif 로 저장

```
dn: dc=samjung,dc=com
objectclass: dcObject
objectclass: organization
o: samjung Company
dc:samjung

dn: cn=manager, dc=samjung, dc=com
objectclass: organizationalRole
cn: manager

dn: ou=people,  dc=samjung, dc=com
ou: people
objectclass: organizationalUnit
objectclass: domainRelatedObject
associatedDomain: samjung.com

dn: ou=contacts,ou=people,  dc=samjung, dc=com
ou: contacts
```

```
ou: people
objectclass: organizationalUnit
objectclass: domainRelatedObject
associatedDomain: samjung.com

dn: ou=group,  dc=samjung, dc=com
ou: group
objectclass: organizationalUnit
objectclass: domainRelatedObject
associatedDomain: samjung.com
```

위에서 ou=contacts 는 아래에서 실제 사용하지는 않으며 이메일주소록을 ldap을 이용할 경우에 사용하면 된다.

```
# ldapadd -x -D 'cn=manager,dc=samjung,dc=com' -W -f top.ldif
Enter LDAP Password:
adding new entry "dc=samjung,dc=com"

adding new entry "cn=manager, dc=samjung, dc=com"

adding new entry "ou=people,  dc=samjung, dc=com"

adding new entry "ou=contacts,ou=people,  dc=samjung, dc=com"

adding new entry "ou=group,  dc=samjung, dc=com"
```

## ldap 프로그램에서의 옵션참고

- ° -w password 로 해도 됨. -W 는 명령행에서 입력
  -x : simple authentication. 기본인증방식임
  -D : binddn 지정
  -f file : 파일에서 입력을 받을 경우 사용
  -W : prompt for simple authentication . 기본인증에서 비밀번호를 별도 입력으로 받을 경우 사용
  -w : 비밀번호를 명령행에서 바로 옵션으로 줌
  -b : searchbase 검색범위 지정

## 위에서 입력한 내용을 검색하기

```
# ldapsearch -x -b 'dc=samjung,dc=com'
version: 2

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

# samjung, com
dn: dc=samjung,dc=com
objectClass: dcObject
objectClass: organization
o: samjung Company
dc: samjung

##...

# search result
search: 2
result: 0 Success

# numResponses: 6
# numEntries: 5
```

참고로 -D 옵션을 이용 binddn을 지정하여 인증을 받고 검색할 수도 있고 -h 옵션을 이용하여 다른 서버에서 검색할 수

있다.

```
ldapsearch -x -b 'dc=samjung,dc=com' -D cn=readonly,ou=admingroup,dc=samjung,dc=com -w
xxxxxxxxxxxx  -h wiki.sds.co.kr
```

# 계정추가하기

## ldap 으로 단일한 리눅스 로그인 만들기

먼저 계정정책을 결정한다. 아래에서는 다음과 같이 하였다고 가정한다.

System accounts : UID < 500
Real people in LDAP : 499 < UID < 10,000
Local users, groups (not in LDAP ) > 10,000

## 로컬 컴퓨터 사용자 엔트리 만들기

ldaptest 라는 계정을 만들며 uid 1000 gid 1000으로 하고 홈디렉토리는 /home/ldaptest 로 함

```
# cat people.ldif
# ldaptest, people, samjung.com
dn: uid=ldaptest,ou=people,dc=samjung,dc=com
cn: ldaptest
objectClass: account
objectClass: posixAccount
objectClass: shadowAccount
objectClass: top
uidNumber: 1000
gidNumber: 1000
homeDirectory: /home/ldaptest
loginShell: /bin/bash
shadowLastChange: 11192
shadowMin: -1
shadowMax: 99999
shadowWarning: 7
shadowInactive: -1
shadowExpire: -1
shadowFlag: 134538308
uid: ldaptest
userPassword: {crypt}$1$OQAQLKrD$ktucNP.aAo/w5gbuAIV6H1
```

아래와 같이 추가하여줌

```
# ldapadd -x -D 'cn=manager,dc=samjung,dc=com' -W -f people.ldif
Enter LDAP Password:
adding new entry "uid=ldaptest,ou=people,dc=samjung,dc=com"
```

아래와 같이 검색함

```
# ldapsearch -x -b "dc=samjung,dc=com" "(objectclass=*)"
```

사용자 지우기

```
ldapdelete -x  -D 'cn=manager,dc=samjung,dc=com'  'uid=ldaptest,ou=people,dc=samjung,dc=com' -W
```

## 기존계정정보 이용하여 마이그레이션하기

/usr/share/openldap/migration/ 디렉토리에 기존의 정보를 마이그레이션하기 위한 프로그램이 있다.
사전에 migrate_common.ph 에서 몇가지 옵션을 수정함. migrate_common.ph 가 변경한 프로그램이고 migrate_common.ph.orig 가 원래의 설정이다.

```
# diff migrate_common.ph migrate_common.ph.orig
71c71
< $DEFAULT_MAIL_DOMAIN = "sds.co.kr";
---
> $DEFAULT_MAIL_DOMAIN = "padl.com";
74c74
< $DEFAULT_BASE = "dc=samjung,dc=com";
---
> $DEFAULT_BASE = "dc=padl,dc=com";
90c90
< $EXTENDED_SCHEMA = 1;
---
> $EXTENDED_SCHEMA = 0;

/usr/share/openldap/migration/migrate_passwd.pl /etc/passwd
/usr/share/openldap/migration/migrate_group.pl /etc/group
```

이 프로그램으로 passwd, gorup 뿐만 아니라 /etc/networks, /etc/protocols, /etc/services, /etc/netgroup 등도 가능하다. 나중에 /etc/hosts 를 LDAP으로 이전하는 곳에서 다시 설명을 한다.

## 그룹 엔트리 만들기

```
# cat group.ldif
dn: cn=webdev,ou=group,dc=samjung,dc=com
objectClass: posixGroup
objectClass: top
cn: webdev
gidNumber: 2000
memberUid: ldaptest

# ldapadd  -x -D 'cn=manager,dc=samjung,dc=com' -W -f group.ldif
Enter LDAP Password:
adding new entry "cn=webdev,ou=group,dc=samjung,dc=com"
```

2000 gid 에 해당하는 webdev 그룹을 만들기 ldaptest 를 이 그룹에 넣어줌

아래와 같이 검색함

1. ldapsearch -x -b 'dc=samjung,dc=com'

# ldap client 설정

## ldap client 설정하기

authconfig 이용하여 설정한다. 이 프로그램을 이용하면 /etc/ldap.conf , /etc/nsswitch.conf, /etc/sysconfig/authconfig, /etc/pam.d/system-auth 파일을 자동으로 바꾸어준다.

User Information Configuration 에서 Use LDAP 선택 -> Next -> Authentication Configuration 에서 Use LDAP Authentication 사용함. Server 및 Base DN에 적당하게 값을 넣음. 여기서는 dc=samjung,dc=com
start_tls 는 나중에 다시 설명한다.

```
# diff /etc/ldap.conf.orig /etc/ldap.conf
18c18
< base dc=example,dc=com
---
> base dc=samjung,dc=com


# diff /etc/openldap/ldap.conf.orig /etc/openldap/ldap.conf
16c16
< BASE dc=example,dc=com
---
> BASE dc=samjung,dc=com

# diff /etc/nsswitch.conf.orig /etc/nsswitch.conf
33,35c33,35
< passwd:     files
< shadow:     files
< group:      files
---
> passwd:     files ldap
> shadow:     files ldap
> group:      files ldap
53c53
< protocols:  files
---
> protocols:  files ldap
55c55
< services:   files
---
> services:   files ldap
57c57
< netgroup:   files
---
> netgroup:   files ldap
61c61
< automount:  files
---
> automount:  files ldap
```

/etc/ldap.conf는 ldap 클라이언트 설정에서 필요한데 몇가지 추가옵션이 있다. 기본설정은 base, hosts 만 바꾸면 작동하는데 아래는 몇가지를 추가하였다. start_tls 를 이용하여 tls 설정, pam_check_host_attr 를 이용하여 사용자별 서버접속제한, pam_filter , pam_login_attribute 를 이용하여 사용자검색시 사용할 objectclass와 login 애트리뷰트를 설정하였다. 또한 nss_base 를 이용하여 해당 정보에 대하여 빠르게 검색할 수 있도록 기본 필터를 설정하였다. 초기 테스팅을 할 경우에는 아래와 같이 옵션을 할 필요는 없다.

```
# grep -v "^#" /etc/ldap.conf
host cent3.tunelinux.pe.kr
base dc=samjung,dc=com
timelimit 120
bind_timelimit 120
idle_timelimit 3600

ssl start_tls
tls_checkpeer yes
tls_cacertfile /etc/openldap/cacerts/cacert.pem

pam_password md5

pam_check_host_attr yes

pam_filter objectclass=posixAccount
pam_login_attribute uid
```

```
nss_base_passwd         ou=people,dc=samjung,dc=com?one
nss_base_shadow         ou=people,dc=samjung,dc=com?one
nss_base_group          ou=group,dc=samjung,dc=com?one
nss_base_hosts          ou=hosts,dc=samjung,dc=com?one
nss_base_netgroup       ou=netgroup,dc=samjung,dc=com?one
```

authconfig 프로그램을 이용해 ldap과 연동을 하면 /etc/sysconfig/authconfig 에서 ldap 부분이 yes로 설정이 바뀐다. (USELDAP, USELDAPAUTH)

```
# grep yes /etc/sysconfig/authconfig
USECRACKLIB=yes
USELDAP=yes
USELDAPAUTH=yes
USEMD5=yes
USESHADOW=yes
```

참고로 ldap 서버를 replication 등을 이용하여 여러대를 사용하는 경우 host 에서 스페이스를 이용해 여러 서버를 지정하면 된다. authconfig에서는 중간에 , 를 이용하여 여러 서버를 지정한다.

```
# grep ^host /etc/ldap.conf
host cent3.tunelinux.pe.kr cent.tunelinux.pe.kr
```

위에서는 anonymous 로 ldap에 접속을 하는데 binddn 과 bindpw 을 이용하여 접근을 제한할 수 있다. 계정정보, 그룹정보를 익명사용자에게 허용할 필요가 없다면 binddn을 이용하여 접근할 사용자를 지정하는 것이 좋을 듯 하다. 이경우 여기서 지정한 binddn이 people 등 필요한 정보에 접근할 수 있도록 접근권한을 설정하고 anonymous 사용자 접속은 막아야 한다.

## group 정보표시

/etc/ldap.conf에 host, base 정보만 넣은 경우 id 등에서 그룹정보가 보이지 않고 숫자로만 나온 경우가 있었다. 이경우 /etc/ldap.conf 에서 바로 위에서 보듯이 nss_base_group 을 설정해주면 되었다.

```
nss_base_group          ou=group,dc=samjung,dc=com?one
```

이러한 정보들은 getent 로 확인해보면 된다. getent passwd, getent group 등으로 확인해보면 된다.

```
# getent passwd
# getent group
```

# 사용자 홈디렉토리 처리

LDAP을 이용하여 사용자 인증을 하는 경우 사용자 LDIF 파일에서 홈디렉토리를 지정한다고 하더라도 실제 디렉토리가 생기지는 않는다. 이에 대한 처리방법은 두가지가 있다.

- autofs 와 nfs를 이용하여 사용자가 로그인할때 nfs에서 자동으로 홈디렉토리 마운트하기 : 사용자 데이터도 동일하게 설정할 경우 편리함.
- pam 의 기능을 이용하여 사용자 홈디렉토리가 없을 경우 자동으로 생성하기 : /etc/pam.d/system-auth 에 다음 모듈을 추가해주면 됨. umask 는 아래에서는 기본 700으로 생성하도록 설정했고 필요에 따라 변경하면 됨

```
session     optional      /lib/security/$ISA/pam_mkhomedir.so skel=/etc/skel umask=0077
```

# /etc/hosts 정보 LDAP에 넣기

/usr/share/openldap/migration/ 에 각종 마이그레이션 도구들이 있다.
migrate_base.pl 는 마이그레이션 가능한 각종 기본정보에 대해서 보여준다.
migrate_base.pl 를 이용하여 hosts 에 대한 기본정보를 뽑고 /etc/hosts 정보를 변환하여 ldap에 넣어준다. 세부설명은 생략하겠다.

```
# ./migrate_base.pl

dn: ou=Hosts,dc=samjung,dc=com
ou: Hosts
objectClass: top
objectClass: organizationalUnit
objectClass: domainRelatedObject
associatedDomain: sds.co.kr
```

위에서 hosts에 해당하는 내용을 ldif 파일로 해서 입력해준다.

migrate_hosts.pl 는 /etc/hosts 정보를 ldif 파일로 바꾸어준다.

```
[migration:root@cent3 migration]# ./migrate_hosts.pl /etc/hosts > hosts.ldif
dn: cn=localhost.localdomain,ou=Hosts,dc=samjung,dc=com
objectClass: top
objectClass: ipHost
objectClass: device
ipHostNumber: 127.0.0.1
cn: localhost.localdomain
cn: localhost

dn: cn=cent3.tunelinux.pe.kr,ou=Hosts,dc=samjung,dc=com
objectClass: top
objectClass: ipHost
objectClass: device
ipHostNumber: 222.112.137.138
cn: cent3.tunelinux.pe.kr
```

```
# ldapadd -x -D 'cn=manager,dc=samjung,dc=com' -W -f hosts.ldif
```

그런후 /etc/nsswitch.conf 를 변경한다.

```
[migration:root@cent3 migration]# grep hosts /etc/nsswitch.conf
#hosts:     db files ldap nis dns
#hosts:      files dns
hosts:      files dns ldap
```

이제 /etc/ldap.conf 에서 hosts 정보를 찾을 수 있도록 정보를 변경한다.

```
[migration:root@cent3 migration]# grep hosts /etc/ldap.conf
# Multiple hosts may be specified, each separated by a
#nss_base_hosts         ou=Hosts,dc=example,dc=com?one
nss_base_hosts          ou=hosts,dc=samjung,dc=com?one
[migration:root@cent3 migration]# getent hosts
```

테스팅과정중에 발견한 중요한 내용이 있다. /etc/nsswitch.conf 에서 hosts 설정순서가 중요하다. ldap 클라이언트에서 자신의 호스트네임을 풀어야한다. 이때문에 dns 항목이 ldap 보다 앞에 오거나 호스트명을 /etc/hosts 파일에 적어주어야 한다. 이렇게 하지 않으면 segmentation fault 에러가 나고 이후부터는 id 등 각종 프로그램에서 계속 세그멘테이션 폴트가 나면서 시스템 작동이 이상해진다.

```
# getent hosts
127.0.0.1       localhost.localdomain localhost
Segmentation fault
```

http://www.mathematik.uni-marburg.de/~gasi/Doc/install/tasks.html#ldap_client_host 참고자료

4.7.5.1 Host Resolving
(2) looping resolver - segmentation fault
The order within /etc/nsswitch.conf is important, and the ldap client code needs to resolve its own hostname!
Therefor dns must be before ldap or the hostname must be in /etc/hosts!

# 서버, 클라이언트 몇가지 옵션

## 서버에서 검색제한하기

slapd.conf 에서 sizelimit , timelimit를 이용하여 검색에 대한 제한을 걸 수 있다.

- sizelimit : 검색요청을 할 경우 클라이언트의 요청에 답하는 최대 엔트리 숫자. 기본값은 500
- timelimit : 검색요청에 응답을 할때 걸리는 최대 시간. 기본값은 3600초(1시간)

## /etc/ldap.conf 주요 옵션에 대하여

/etc/ldap.conf 주요 옵션은 다음과 같다.

- host는 ldap 서버, base 는 base dn이다.
- ssl start_tls
  는 TLS를 사용하는 경우 체크하는 옵션이다. 암호화되어 통신하는 것이다.
- pam_check_host_attr 는 hosts를 이용하여 사용자별로 접속할 호스트를 제한하는데 사용한다.
- pam_filter 는 사용자 인증시 사용할 필터이다. pam_login_attribute 는 사용자의 로그인 명과 일치하는 attribute를 지정한다.
- nss_base_xxx 는 nss_ldap 에서 검색하는 부분을 지정하여 LDAP 서버의 부하를 줄일 수 있다. passwd, shadow는 상관없지만 group, hosts는 등록을 해주어야했다.

```
ssl start_tls
tls_cacertdir /etc/openldap/cacerts
pam_password md5
pam_check_host_attr yes

pam_filter objectclass=posixAccount
pam_login_attribute uid

nss_base_passwd         ou=people,dc=samjung,dc=com?one
nss_base_shadow         ou=people,dc=samjung,dc=com?one
nss_base_group          ou=group,dc=samjung,dc=com?one
nss_base_hosts          ou=hosts,dc=samjung,dc=com?one
```

# 호스트, 사용자별 접근제한

특정 호스트, 사용자를 지정하여 접근을 제한할 수 있는데 두가지 방법이 있다.
첫번째는 특정한 호스트에 접속가능한 사용자들을 지정하는 방식(a.server 에 a,b,c 사용자 접속가능)이 있고 두번째는 특정한 사용자가 접속가능한 호스트들을 지정하는 방식(a 사용자는 가,나,다 서버에 접속가능)이 있다.
실제 사용하는 경우 뒤의 방식이 더 편리하다. 앞의 방식은 클라이언트에서 설정을 일일이 세팅해야하지만 뒤의 방식은 클라이언트에서 동일한 설정을 유지하되 ldap서버에서 변경을 할 수가 있다.

## 특정 사용자가 접속하능한 호스트 설정하기

/etc/ldap.conf 에서 pam_check_host_attr yes로 해줌. /etc/openldap/ldap.conf가 아니다.
사용자를 추가할때 host 에 접속가능한 호스트 지정. 여기서 IP로 지정하면 접속이 되지 않았고 정확한 도메인명을 지정해야한다.

```
# test, people, samjung.com
dn: uid=test,ou=people,dc=samjung,dc=com
#### ##
host: kldp.org
host: cent3.tunelinux.pe.kr
```

pam 설정은 변경할 필요가 없다.

## 특정 호스트에 접속가능한 사용자 제한하기

ou=hosts 가 먼저 있어야 한다.

```
# cat host.ldif
dn: ou=hosts,  dc=samjung, dc=com
ou: hosts
objectclass: organizationalUnit
objectclass: domainRelatedObject
associatedDomain: samjung.com

# ldapadd -x -D 'cn=manager,dc=samjung,dc=com' -W -f host.ldif
```

이제 특정 호스트와 사용자에 대한 정보를 입력한다. 아래에서는 cn을 linux 를 하였다.

```
# cat iphost.ldif
dn: cn=linux,ou=hosts,dc=samjung,dc=com
objectClass: ipHost
objectClass: device
objectClass: extensibleObject
ipHostNumber: 192.168.0.23
cn: linux.samjung.com
cn: linux
member: uid=test,ou=people,dc=samjung,dc=com
member: uid=test2,ou=people,dc=samjung,dc=com

# ldapadd -x -D 'cn=manager,dc=samjung,dc=com' -W -f iphost.ldif
```

위에서는 192.168.0.23 에 test, test2 계정만 접속가능하도록 설정하였다.

ldap에 위의 정보를 입력한 후 각 ldap client 에 위 기능을 사용할 수 있도록 설정해야 한다.

이는 /etc/ldap.conf 에 다음 항목을 추가한다. 위에서 사용한 dn을 넣어주어야 한다.

```
pam_groupdn cn=linux,ou=hostss,dc=samjung,dc=com
pam_member_attribute member
```

테스팅을 한 결과 /etc/ldap.conf 에 pam_groupdn 설정을 두개 넣으면 작동을 하지 않았다.
그렇지만 각 ldap client 쪽에 이 설정이 두가지 들어갈 일이 없으므로 문제가 되지는 않는다.

iphost.ldif 에 설정한 내용을 각 ldap client 별로 ldap에 넣어주고 이후에는 그 설정내용만 계속 수정하면 된다.

# NIS netgroup 사용하여 사용자, 호스트별 접근제한하기

## 관련자료

오렐리 LDAP admin 117쪽
페도라 디렉토리 서버 위키의 문서중 "System Access Control using LDAP backed NIS netgroup"
http://directory.fedora.redhat.com/wiki/Howto:netgroup

## NIS netgroup 기능

NIS는 Sun에서 나온 기술로 여러대의 시스템을 통합적으로 관리하기 위해 나왔다. 사용자계정, 그룹, /etc/hosts 등을 통합해서 관리할 수 있다.
NIS netgroup은 다음과 같은 기능을 제공한다.

- 개별 시스템 또는 시스템그룹에 사용자와 그룹 로그인 접근 제어
- NFS 접근 제어 목록 관리
- 사용자,그룹에 대한 sudo 명령어 접근제어
- dsh(distributed shell)을 이용하여 원격 명령 실행 또는 시스템그룹에 작업
- cfengine을 이용하여 정책 기반의 시스템 설정관리

tcp 래퍼를 통하여 간단한 예를 살펴보자.

```
# /etc/hosts.deny
sshd: ALL
# /etc/hosts.allow
sshd: @sysadmin
```

위에서 sysadmin netgroup는 다음과 같이 개별 호스트로 구성할 수 있다.

```
sysadmin (a.com,-,-)(b.com,-,-)
```

또는 다른 netgroup을 포함할 수 있다.

```
all_sysadmin sysadmin secure_clients
```

(a.com,-,-) 구성은 host, user, NIS-domain 으로 구성이 되며 -는 생략을 해도 된다. 마지막 NIS-domain은 생략을 해도 LDAP과 cfengine 에서 사용이 가능하였다.

이를 이용하면 시스템그룹별, 사용자그룹별로 여러가지 작업을 제어할 수 있고 시스템그룹과 사용자그룹의 조합도 가능하다.

## LDAP 에서 netgroup 구현

LDAP에서는 structural nisNetgroup 오브젝트 클래스를 이용하여 netgroup 기능을 구현할 수 있다.

nisNetgroup 오브젝트 클래스에서 rdn은 cn을 쓰며두가지 중요한 attributes 가 있다.

nisNetgroupTriple : 사용자(,love,samjung.com), 시스템 (cent.tunelinux.pe.kr,,samjung.com) 을 지정할 수 있으며 여러개의 값이 들어갈 수 있다.
memberNisNetgroup : 다른 netgroup 를 포함할 수 있다. 대그룹, 소그룹 등으로 분류하여 편리하게 사용할 수 있는 기능이다. 이또한 여러개의 값을 가질 수 있다.

먼저 ou를 생성한다. LDIF 파일로 저장하여 ldapadd로 넣으면 된다.

```
dn: ou=netgroup,dc=samjung,dc=com
objectClass: organizationalUnit
ou: netgroup

dn: cn=sysadmin,ou=netgroup,dc=samjung,dc=com
objectClass: nisNetgroup
objectClass: top
cn: sysadmin
description: netgroup test group
nisNetgroupTriple: (cent1.tunelinux.pe.kr,-,-)
nisNetgroupTriple: (cent2.tunelinux.pe.kr,-,-)

dn: cn=sysadmin2,ou=netgroup,dc=samjung,dc=com
objectClass: nisNetgroup
objectClass: top
cn: sysadmin2
description: netgroup test group2
memberNisNetgroup: sysadmin
memberNisNetgroup: sysadmin2

dn: cn=allusers,ou=Netgroup,dc=samjung,dc=com
objectClass: nisNetgroup
objectClass: top
cn: users0
nisNetgroupTriple: (,a,)
nisNetgroupTriple: (,b,)
description: All QA users in my organization
```

sysadmin은 host가 cent1.tunelinux.pe.kr, cent2.tunelinux.pe.kr 를 넷그룹으로 묶으며 sysadmin2는 memberNisNetgroup을 이용하여 sysadmin, sysadmin2 넷그룹을 묶는 것이다.
nisNetgroupTriple 과 memberNisNetgroup은 같이 들어갈 수도 있다.
alluser는 a,b 사용자를 묶었다. 위에서 설명한바와 같이 NIS 도메인 명은 입력을 하지 않아도 작동하는데는 문제가 없었다.
페도라 디렉토리 서버 위키의 문서중 "System Access Control using LDAP backed NIS netgroup"에는 다음과 같이 나와있다.
http://directory.fedora.redhat.com/wiki/Howto:netgroup

```
Finally to enable the netgroup query, NISDOMAIN must be defined (in /etc/sysconfig/network)
even though it is not used. This is required because the innetgr() call is used and it requires
a nisdomainname as a paramter. Once the functions resolves to LDAP via nsswitch.conf, the
nisdomainname in no longer required.
```

필요한 엔트리를 추가한 후 /etc/ldap.conf 에서 netgroup 검색을 위하여 nss_base_netgroup 을 추가한다.

```
nss_base_netgroup       ou=netgroup,dc=samjung,dc=com?one
```

OS에서 netgroup을 찾을 수 있도록 /etc/nsswitch.conf 에서 netgroup 에 대한 설정을 한다.

```
netgroup:   ldap
```

getent 프로그램을 이용하여 위에서 입력한 netgroup을 검색해본다.

```
# getent netgroup sysadmin
sysadmin               (cent1.tunelinux.pe.kr, , ) (cent2.tunelinux.pe.kr, , )
```

이러한 설정을 이용하여 위에서 sshd는 sysadmin 에 속한 호스트에서만 접속을 하도록 설정을 할 수 있는 것이다.

## PAM 접근제어 연동

tcp 래퍼만이 아니라 넷그룹을 이용하여 PAM 의 접근권한 제어와 연관을 시킬 수가 있다.
이에 대한 내용은 페도라 디렉토리 서버의 위키에 자세히 나와있다.

위와 같은 작업을 하여 특정 호스트와 특정 사용자별로 그룹을 묶는다.
bobby, joey 사용자를 QAUsers 그룹으로 만든다.

```
dn: cn=QAUsers,ou=Netgroup,dc=example,dc=com
objectClass: nisNetgroup
objectClass: top
cn: QAUsers
nisNetgroupTriple: (,bobby,example.com)
nisNetgroupTriple: (,joey,example.com)
description: All QA users in my organization
```

qa01, qa02 호스트를 QASystems 그룹으로 만든다.

```
dn: cn=QASystems,ou=Netgroup,dc=example,dc=com
objectClass: nisNetgroup
objectClass: top
cn: QASystems
nisNetgroupTriple: (qa01,,example.com)
nisNetgroupTriple: (qa02,,example.com)
description: All QA systems on our network
```

PAM 에서 /etc/security/access.conf 파일을 이용하여 ip 에 따라 접속가능한 호스트와 사용자를 지정할 수 있다.
이에 대해서는 별도로 PAM 정보를 참고한다.
access.conf 파일에서 nis의 넷그룹은 @netgroupname 형태로 이용하면 된다. 여기서 호스트명이나 사용자명 한가지만 이용하는 것이 아니라 두가지를 결합하면 여러가지 편리한 점이 있다.
아래의 내용은 10.x.x.x 네트워크에서 QASystems에 QAUsers 가 접속할 수 있도록 하는 것이다.

```
+ : @QAUsers@@QASystems : 10.
```

아래의 경우는 root 사용자는 로컬에서만 접속하고 Admins 넷그룹은 10.x 네트워크에서 접속할 수 있도록 하며 나머지는 모두 막는 설정이다.

```
+ : root : LOCAL
+ : @Admins : 10.
- : ALL : ALL
```

## cfengine 에서의 사용

cfengine은 각종 시스템작업을 자동화할 수 있는 프로그램이며 별도 자료를 참고하기 바란다.
http://www.cfengine.org/docs/cfengine-Reference.html#groups
NIS netgroup을 이용하는 경우에는 +나 +@ 기호를 이용한다. 여기서 유용한 것이 netgroup except 이다. 아래에서 testgroup은 mynetgoup을 포함하고 있는데 mynetgoup 에서 특정 호스트만 빼려고 할 경우에는 - 기호를 이용하여 지정하면 된다.

```
groups:
        science = ( +science-allhosts )
        physics = ( +physics-allhosts )
        physics_theory = ( +@physics-theory-sun4 dirac feynman schwinger )
        testgroup = ( +mynetgroup -specialhost -otherhost )
```

## 참고사항

### host 이름에 대하여

dns에 등록되어있지 않아도 ldap의 hosts 에 들어가있으면 동일하게 동작한다.

### nisNetgroupTriple 추가, 변경시

실제 사용하면서 문제가 부딪힌 것이 있다. nisNetgroupTriple 을 추가하려고 하는 경우에는 additional info: modify/add: nisNetgroupTriple: no equality matching rule 라는 에러가 난다.
attribute 정의에서 nisNetgroupTriple 은 매칭 룰이 없다. 이 부분이 영향을 미치는 것 같다.
좋은 방법은 아닌듯하지만 스키마에서 EQUALITY 와 SYNTAX를 수정해주었지만 제대로 작동하지는 않았다.

```
# cat mod.txt
dn: cn=sysadmin2,ou=netgroup,dc=samjung,dc=com
changetype: modify
add: nisNetgroupTriple
nisNetgroupTriple: (cent2.tunelinux.pe.kr,,)

#  ldapmodify -D "cn=manager,dc=samjung,dc=com" -W -x -v -f mod.txt
ldap_initialize( <DEFAULT> )
add nisNetgroupTriple:
        (cent2.tunelinux.pe.kr,,)
modifying entry "cn=sysadmin2,ou=netgroup,dc=samjung,dc=com"
modify complete
ldap_modify: Inappropriate matching (18)
        additional info: modify/add: nisNetgroupTriple: no equality matching rule
```

nisNetgroupTripple attibutetype

```
attributetype ( 1.3.6.1.1.1.1.13 NAME 'memberNisNetgroup'
        EQUALITY caseExactIA5Match
        SUBSTR caseExactIA5SubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 )

attributetype ( 1.3.6.1.1.1.1.14 NAME 'nisNetgroupTriple'
        DESC 'Netgroup triple'
```

```
        SYNTAX 1.3.6.1.1.1.0.0 )
```

nisNetgroupTriple은 초기 한개 입력가능하며 한개만 있을 경우 수정, 삭제가 가능한데 두개이상 추가가 되지 않는다.
매칭룰때문에 생기는 문제라고 판단이 되며 이럴 경우 해당 dn을 삭제하고 신규로 dn를 넣어주어야 한다.

## 사용자 접근제한 어떤 방법이 좋을까?

호스트별로 접속가능한 사용자를 지정하는 방식은 접속하려는 클라이언트 설정이 모두 달라지므로 불편하다.
(pam_groupdn, pam_member_attribute 설정)
pam_check_host_attr 또는 LDAP에 NIS를 연동하는 방식이 관리상 편리할 것이다.

각자의 장단점을 생각해보자.
pam_check_host_attr 을 이용하면 각 사용자별로 접속할 수 있는 호스트를 지정한다.
모든 것을 LDAP에서 관리하고 /etc/ldap.conf 에서 pam_check_host_attr 지정하는 것 외에 별도의 설정이 필요없으므로 구성이 간단하다.
하지만 시스템과 사용자규모가 커지면 별도의 관리툴을 만들지 않으면 불편하다.

NIS를 이용하는 경우에는 설정은 좀더 복잡해지지만 사용자, 시스템별로 그룹을 만들고 이 그룹을 필요에 따라 조정할 수 있다.
/etc/security/access.conf는 시스템에 달라지는것이 아니라 모든 시스템에서 동일한 내용을 공유할 수 있다.
기본설정은 동일하되 특정 그룹에 대한 조정은 ldap을 통하여 하면 된다.
한가지 단점이라면 nisNetgroupTriple은 한개만 입력가능, 한개만 있을 경우 수정, 삭제가 가능한데 두개이상 추가는 되지가 않는다.
매칭룰때문에 생기는 문제라고 판단이 되며 이럴 경우 해당 dn을 삭제하고 신규로 dn를 넣어주어야 한다.
이러한 불편함은 있지만 기본 제공되는 기능만으로 가장 강력하게 접근제어를 할 수가 있다.
또한 NIS기능을 cfengine 등 다른 프로그램에서도 활용이 가능하다.

# user 변경 프로그램 - cpu

passwd 프로그램을 이용해서 사용자를 변경하여도 된다. 그렇지만 사용자 생성은 ldif 파일로 직접 넣거나 cpu 프로그램 이용 또는 ldap 관리자툴을 이용해야 한다. cpu가 사용자 계정 및 그룹관리에 편리하다.

http://cpu.sourceforge.net/ 최신버전 다운로드

- rpmfind 에서 cpu rpm을 다운로드 받아도 됨. rhel4 버전에 맞춘 rpm이 있음.
  여기서 설치한 rpm의 cpu 프로그램은 다른 사용자도 사용할 수 있으므로 root만 사용하도록 조정한다.

```
[migration:root@cent3 migration]# ll /usr/sbin/cpu
-rwxr-xr-x  1 root root 12127 Feb 17  2005 /usr/sbin/cpu
[migration:root@cent3 migration]# chmod 700 /usr/sbin/cpu
```

openldap-devel 필요함

```
./configure --prefix=/usr/local/cpu
make
make install
```

이제 /usr/local/cpu 에 프로그램이 설치가 된다.

```
# grep samjung /usr/local/cpu/etc/cpu.conf
BIND_DN         = cn=Manager,dc=samjung,dc=com
USER_BASE       = ou=People,dc=samjung,dc=com
GROUP_BASE      = ou=Group,dc=samjung,dc=com
```

위와 같이 dn을 바꾸어준다.

```
#HASH = "md5"
HASH = "crypt"
```

HASH 를 md5 에서 crypt 로 바꾸어준다.

여기서 sldapd.conf 의 root 비밀번호를 넣어주어야 한다.

```
BIND_PASS       = xxxx


MAX_UIDNUMBER = 10000
MIN_UIDNUMBER = 1000
MAX_GIDNUMBER = 10000
MIN_GIDNUMBER = 1000
```

MIN_UIDNUMBER, MIN_GIDNUMBER 를 100에서 적절한 값으로 바꾼다.

```
# /usr/local/cpu/sbin/cpu useradd test
# /usr/local/cpu/sbin/cpu userdel test
$ /usr/local/cpu/sbin/cpu usermod -p  test2

[migration:root@localhost openldap]# id test
uid=1001(test) gid=1001(test) groups=1001(test)
[migration:root@localhost openldap]# /usr/local/cpu/sbin/cpu groupmod -g 1005 test
Group test successfully modified!
[migration:root@localhost openldap]# id test
uid=1001(test) gid=1001 groups=1001,1005(test)
[migration:root@localhost openldap]# /usr/local/cpu/sbin/cpu groupmod -n test222 test
Group test222 successfully modified!
[migration:root@localhost openldap]# id test
uid=1001(test) gid=1001 groups=1001,1005(test222)
```

편하게 사용을 하려면 path에 추가해주면 좋다.

```
export PATH=$PATH:/usr/local/cpu/sbin
export MANPATH=$MANPATH:/usr/local/cpu/man
man cpu-ldap
```

cpu cat 은 전체 사용자, 그룹을 본다.

```
[migration:root@cent ~]# cpu cat
User Accounts
ldaptest:x:1001:1001::/home/ldaptest:/bin/bash
ldap2:x:1000:1002::/home/ldap2:/bin/bash

Group Entries
webdev:x:2000:
test:x:1000:
ldaptest:x:1001:
ldap2:x:1002:
```

사용자 패스워드 변경한다.

```
[migration:root@cent ~]# cpu usermod -p ldaptest
```

관리를 위해서는 먼저 필요한 그룹을 생성하고 그 사용자를 추가해주는 것이 좋을 것이다. 기본값은 사용자를 생성시 동일한 이름의 그룹을 생성한다. 그러므로 처음 생성시 -g 옵션을 이용하여 그룹을 지정하는것이 좋다. 아니면 사용자 생성후 그룹을 바꾸어주어도 된다.

```
[migration:root@cent3 openldap]# cpu useradd -g test5 ilove
[migration:root@cent3 openldap]# cpu usermod -g test ilove
```

사용자를 추가할때 -p옵션뒤에 패스워드를 넣어줄 수 있다.

USERGROUPS = no
USERS_GID = 10000

cpu.conf 에서 USERGROUPS 은 기본 yes로 되어있다. 사용자 생성시 같은 이름의 gid로 그룹을 생성한다. 이를 no로 바꾸어주고 USERS_GID 에 특정 gid를 넣어주면 자동으로 해당 gid로 그룹을 생성한다. 일관된 관리를 할때 편리할 것이다.

# nfs, autofs 세팅

nfs, autofs는 홈디렉토리를 사용자가 로그인시 자동으로 파일서버에서 마운트하는 경우에만 사용하면 됩니다.
장점은 홈디렉토리를 각 서버마다 만들지 않고 일괄적으로 관리할 수 있다는 것이다.
여기에 쿼터설정하여 홈디렉토리가 계속 커지는 것을 막도록 하는 것이 좋을 것이다.

## nfs 서버 세팅

nfs의 경우 방화벽 설정을 하는 경우에 해당 port만 여는 것이 아니라 작업이 필요하다.
이에 대해서는 nfs 방화벽설정에 대한 내용을 참고한다.
그리고 portmap, nfs 서버데몬을 시작시 작동하도록 해야 할 것이다.

```
# cat /etc/exports
/tmp   192.168.0.0/255.255.255.0(rw,sync)

# /etc/init.d/nfs start
```

## autofs 세팅

autofs 세팅은 접속할 대상 서버에서 모두 작업을 해야 한다.
ldap을 이용하기전에 먼저 autofs를 먼저 테스팅을 해보기 바란다.
시작시 autofs를 자동으로 시작하도록 설정한다.

auto.master 파일이 메인파일이며 여기에서 마운트 포인트와 세부 설정파일을 지정함.
아래에서는 /home 디렉토리에 접근하는 경우 /etc/auto.home 파일을 참고하며 auto.home 은 /home 의 모든 하위 디렉토리에 접근하는 경우 nfs 192.168.0.24:/tmp 의 해당 디렉토리에 마운트함

```
# cat /etc/auto.master
/home   /etc/auto.home --timeout=5

# cat /etc/auto.home
*               -rw,soft,intr           192.168.0.24:/tmp/&
```

home 디렉토리 공유하기 위해 automount 세팅하기 (사전에 autofs 는 세팅을 해야함)

> autofs 스키마 설정
>
> autofs를 이용할 경우 autofs.schema 을 sldapd.conf에 추가해주어야 한다.
> 그리고 autofs 설정에 대한 acl도 설정을 해야한다.
> include /etc/openldap/schema/redhat/autofs.schema

```
# cat auto.master.ldif
dn: ou=auto.master,dc=samjung,dc=com
objectClass: top
objectClass: automountMap
ou: auto.master

dn: cn=/home,ou=auto.master,dc=samjung,dc=com
objectClass: automount
cn: /home
automountInformation: ldap:ou=auto.home,dc=samjung,dc=com

# ldapadd -x -D 'cn=manager,dc=samjung,dc=com' -W -f auto.master.ldif
Enter LDAP Password:
adding new entry "ou=auto.master,dc=samjung,dc=com"

adding new entry "cn=/home,ou=auto.master,dc=samjung,dc=com"

# cat auto.home.ldifc
dn: ou=auto.home,dc=samjung,dc=com
objectClass: top
objectClass: automountMap
ou: auto.home

dn: cn=*,ou=auto.home,dc=samjung,dc=com
objectClass: automount
cn:*
automountInformation:   192.168.0.24:/tmp/&

# ldapadd -x -D 'cn=manager,dc=samjung,dc=com' -W -f auto.home.ldifc
Enter LDAP Password:
adding new entry "ou=auto.home,dc=samjung,dc=com"

adding new entry "cn=test,ou=auto.home,dc=samjung,dc=com"
```

이렇게 하는 경우 /etc/auto.master 를 ldap 에서 사용할 수 있도록 바꾸어 줄수 있다.
ldap기반에 autofs 이용시 /etc/auto.master 파일만 설정하면 되고 시작시 autofs만 시작하도록 하면 된다.
참고로 timeout은 해당 디렉토리를 자동으로 마운트하고 사용하지 않을 경우 어느정도의 시간을 대기하다록 할 것인지를 지정한다.
해당디렉토리에 자주 접속하는 것이 아니라면 시간은 더 짧게 해놓아도 될 것이다.

```
# cat /etc/auto.master
#/home  /etc/auto.home --timeout=5
/home   ldap:192.168.0.23:ou=auto.home,dc=samjung,dc=com --timeout=5
```

# 각종 애플리케이션 LDAP 연동

## 참고사항

- outlook express에서는 고급검색에서 메일이름을 가지고 검색할 수 있어서 특정 도메인에 대한 메일을 볼 수 있어 편리하다. 그러나 Microsoft Outlook 는 사용자이름이나 id만 가지고 검색할 수 있다. 이는 검색필터가 다르기 때문이다. outlook은 회사명으로 검색을 할 경우 company 라는 attr을 찾는다. 그래서 company attr을 사용자 정의해주고 이를 이용하여 검색할 수 있다. 이에 대한 자료는 http://database.sarang.net/index.php?inc=read&aid=2318&criteria=ldap&subcrit=tutorials DSN의 자료를 참고한다.
- text plain이 아니라 암호화하여 주소록 검색을 하려면 보안인증(ssl)을 사용하면 된다. LDAP서버에서 만든 CA인증서를 .cer 등으로 복사하여 윈도우에서 인증서을 가져오면 사용할 수 있다.

## outlook 등 이메일클라이언트 세팅하기

위에서 ou=people,dc=samjung,dc=com 에 입력한 사용자정보는 아웃룩, 선더버드 등의 주소록에서 활용을 할 수 있다.

### 아웃룩

outlook express 에서는 도구->계정 으로 가서 디렉토리 서비스를 선택한다.
디렉토리 서비스 계정에 적절한 이름을 택하여 찾기 쉽도록 넣는다.
서버 이름에 ldap 서버 정보를 입력한다.
로그인 필요에서는 위에서 만든 ldaptest 등을 이용하면 된다. uid=ldaptest,ou=people,dc=samjung,dc=com 를 넣어주면 될 것이다.
암호는 위 id에 해당하는 비밀번호를 넣으면 된다.
보안 암호 인증을 사용하여 로그인은 잘 모르겠다.
고급에서 검색기준을 입력한다. ou=people,dc=samjung,dc=com

이제 outlook express 에서 주소 -> 사람찾기를 선택하여 ldap 디렉토리를 지정하고 검색조건을 입력하면 된다. 그런데 여기서 이름이나 메일을 입력하면 mail, cn, sn, givenname 에서 입력한 단어로 시작한 것을 찾는다. 그런데 고급조건을 고르면 여러가지 조건을 선택할 수 있고 전자메일에 특정 도메인을 포함하여 검색하면 한 회사의 메일을 빠르게 검색할 수 있다.

Microsoft Outlook 에서는 다음과 같다.
도구->전자메일계정-> 디렉토리 : 새 디렉토리 또는 주소록 추가

서버이름 : intranet.sds.co.kr
로그온정보 : 로그온 필요에 표시함. 사용자이름에 uid=joon,ou=people,dc=sds,dc=co,dc=kr 형태로 입력하며 uid에는 자신의 id를 입력함.
기타설정->검색에서 검색기준을 지정하는데 ou=people,dc=sds,dc=co,dc=kr 하면 된다.
도구메뉴의 찾기에서 검색을 하여 사용하면 된다.

### 선더버드

선더버드에서는 계정설정->주소->디렉토리 편집에서 디렉토리 서비스를 추가한다.
이름은 적절한 이름을 택하여 찾기 쉽도록 넣는다.
호스트 이름에 ldap 서버 정보를 입력한다.
기본 dn에 ou=people,dc=samjung,dc=com 를 입력한다. 기준이 되는 dn을 입력하는 것이다.
포트번호는 ldap 포트번호를 적는다.
DN 바인드는 인증을 사용할 경우에 해당한다.
uid=ldaptest,ou=people,dc=samjung,dc=com
암호는 접속시 입력을 하면 된다.

### 참고사항

현재 기본설정은 다른 사용자도 read 권한을 주기때문에 아웃룩에서 로그인필요, 선더버드에서 DN 바인드를 선택하지 않는다고 하더라도 주소록 검색이 가능하다.
이 부분은 ldap 서버 설정에서 acl을 주어야 할 것이다.

참고로 이메일클라이언트는 읽기 전용이다. 또 검색을 해서 이용해야하는 불편이 있다.

## 웹주소록 프로그램

- /usr/share/doc/labe-3.3/REAME 파일을 참고.
  여기서 먼저 suffix, rootdn를 만들어주고 ldap 대몬을 다시 띄움. 아래 스키마 추가도 여기에서 언급하고 있음.
- http://sourceforge.net/projects/labe/ 여기에서 다운로드 받아 설치하면 된다.
  설정은 ldap을 이해하고 있으면 간단하다. rpm으로 설치하면 /var/www/html/labe/ 디렉토리에 웹프로그램설치가 되고 setup.sh 에서 적절한 답변을 해주면 된다.
  참고로 이유는 모르겠는데 /etc/openldap/slapd.conf 에서 labe 프로그램이 사용하는 스키마를 수동으로 추가해준다. 이는 자동으로 되지 않는 듯하다.

```
include /etc/openldap/schema/extension.schema
```

  /etc/labe/connect.conf 파일이 ldap 접속에 대한 설정파일이며 여기에 서버주소, port, bind, rootdn 정보가 들어간다. 이는 위의 스크립트를 실행하면 생성이 되는 것이다.

## 웹주소록 ACL 설정으로 인증된 사용자만 읽도록 하기

아래와 같이 기본 권한을 none으로 주고 users (dn이 존재하고 패스워드를 제시한 사용자)에게만 read 권한을 주는 것으로 바꾸니 인증을 해야 접속이 된다. ACL 설정부분은 추후에 좀더 살펴봐야함

```
access to attr=userPassword
        by self write
        by anonymous auth
        by dn="cn=manager,dc=samjung,dc=com"        write
        by* compare
access to*
        by self write
        by dn="cn=manager,dc=samjung,dc=com"        write
        by users read
```

위에서 users 에 read 권한을 주지 않으면 다른 정보도 볼수가 없다.

defaultaccess none 가 오렐리 책등에서는 나오는데 openldap 버전이 올라가면서 기본적으로 aci가 설정되지 않으면 거부로 동작이 바뀐듯하다.

# 아파치 인증에 LDAP 사용하기

- 연동방법만 간략히 설명
- http://httpd.apache.org/docs/2.0/ko/mod/mod_auth_ldap.html apache 에서 ldap 인증 아파치 공식한글문서중 관련내용
- htaccess 에서 아래와 같이 사용하면 됨. dc=samjung,dc=co 이 부분을 적절히 바꾸면 될것임. user, group, dn, ldap attribute 등을 이용하여 접속을 제한할 수 있다.
- mod_auth_ldap 에서는 두가지 과정을 거친다. 먼저 사용자를 인증한다. (search/bind) 그러고나서 인증받은 사용자가 요청한 정보에 접근을 허용받았는지를 확인한다 (compare)

```
[migration:joon@localhost moniwiki]$ cat .htaccess
```

```
AuthType Basic
AuthName "joon wiki system"
AuthLDAPURL ldap://localhost:389/ou=people,dc=samjung,dc=com?uid?sub?(objectClass=*)
require valid-user
```

- 만약 ACL에 익명사용자가 사용자정보에 접근을 하지 못하도록 해놓았다면 정보를 검색할 수 있는 binddn을 지정해주어야 한다. 아래와 같은 형태로 위에 추가를 해주어야 한다.

```
AuthLDAPBindDN cn=readonly,dc=samjung,dc=com
AuthLDAPBindPassword readpassword
```

## samba, ldap 연동

구글검색해서 http://aput.net/~jheiss/samba/ldap.shtml 사이트를 보고했지만 잘 되지 않았음. 시간걸릴듯하여 그냥 넘어갔음

# ldap 에서 TLS 사용한 암호화 통신

참고자료
http://www.openldap.org/doc/admin23/tls.html
Centralize user accounts with OpenLDAP http://www-128.ibm.com/developerworks/library/l-openldap/index.html
SSL 참고자료 http://wiki.kldp.org/wiki.php/DocbookSgml/SSL-Certificates-HOWTO

아래 내용은 자체적으로 ldap만 신경쓸 경우 사용하면 되지만 인증서를 한군데에서만 사용하는것이 아니기때문에 "SSL 인증서 만들어서 여러 서비스에 활용하기"의 내용을 참고해서 사용하는 것이 좋다.

## 인증 메커니즘

LDAPv3 에서는 클라이언트 인증에 여러가지 메카니즘을 사용한다.

- anonymous authentication
- simple authentication
- simple authentication over SSL/TLS
- simple authentication and Security Layer (SASL)

SSL/TLS는 두가지 방법이 있다. ssl을 통해 ldap을 사용하는 방법(ldaps, tcp port 636)보다는 StartTLS LDAP 확장기능으로 사용하는 것이 좋다. StartTLS 는 tcp 389 port(ldap포트)를 통해서 TLS 통신을 할 수 있는 기능이다. 서버의 같은 포트에서 클라이언트의 요청에 따라 암호화된 세션과 암호화되지 않은 세션을 모두 처리할 수 있다.

## 인증서 생성

root CA 가 없을 경우 먼저 생성을 해준다. 해당 정보는 시스템에 맞게 적절하게 수정을 한다. Common Name은 해당 서버의 호스트명을 지정한다.
참고로 openssl 기본설정(/usr/share/ssl/openssl.cnf )은 인증서의 유효기간이 default_days 옵션으로 365일이다.
아래 CA 스크립트에서도 DAYS 옵션을 이용하여 365일로 지정하고 있다.
openssl에 대해서는 별도의 자료를 참고하기 바란다.

```
# cd /usr/share/ssl/misc
# ./CA -newca
CA certificate filename (or enter to create)

Making CA certificate ...
Generating a 1024 bit RSA private key
.............................................................++++++
.++++++
writing new private key to './demoCA/private/./cakey.pem'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
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) [migration:KO]:
State or Province Name (full name) [migration:gurogu]:
Locality Name (eg, city) [migration:seoul]:
Organization Name (eg, company) [migration:Samjung dataservice]:
Organizational Unit Name (eg, section) [migration:ITservice]:
Common Name (eg, your name or your server's hostname) [migration:cent3.tunelinux.pe.kr]:
Email Address [migration:joon@sds.co.kr]:
```

이제 LDAP서버에서 사용할 서버 인증요청서(CSR)을 생성한다.
개인키는 slapd-key.pem 으로 지정하고 slapd-req.pem 이 CSR이다. 여기서 nodes 옵션을 쓴것은 ldap서버를 내리고 올려줄때 비밀번호를 넣어주지 않도록 하기 위해서이다.

```
openssl req -new -nodes  -keyout slapd-key.pem -out slapd-req.pem -days 1825
```

이제 앞에서 생성한 root CA로 인증서 사인을 한다.

```
openssl ca -out slapd-cert.pem -infiles slapd-req.pem
```

위에서 생성한 인증서를 적절한 디렉토리로 옮긴다. 참고로 CA키는 /etc/openldap/cacerts 에 두는데 CA 키 말고 아래에서 slapdcert.pem 도 이 디렉토리에 두면 TLS 기능이 제대로 작동하지 않는다. 이 디렉토리에서 ca 키를 찾도록 해 놓아서 에러가 나는 듯하다. 자세한 이유까지는 모르지만 다른 디렉토리에 두면 되므로 주의만 하면 될 것이다.

```
# cp -p slapd-key.pem /etc/openldap/slapdkey.pem -> private key
# cp -p slapd-cert.pem /etc/openldap/slapdcert.pem -> certificate
# chown ldap:ldap /etc/openldap/slapdcert.pem
# chmod 644 /etc/openldap/slapdcert.pem
# chown ldap:ldap /etc/openldap/slapdkey.pem
# chmod 400 /etc/openldap/slapdkey.pem

# cp /usr/share/ssl/misc/demoCA/cacert.pem /etc/openldap/cacerts/cacert.pem -> CA certificate
# chown ldap:ldap /etc/openldap/cacerts/cacert.pem
# chmod 644 /etc/openldap/cacerts/cacert.pem

### ##### #### ## ### #####.
cp -f slapd-key.pem /etc/openldap/slapdkey.pem
cp -f slapd-cert.pem /etc/openldap/slapdcert.pem
chown ldap:ldap /etc/openldap/slapdcert.pem
chmod 644 /etc/openldap/slapdcert.pem
chown ldap:ldap /etc/openldap/slapdkey.pem
chmod 400 /etc/openldap/slapdkey.pem

cp -f /usr/share/ssl/misc/demoCA/cacert.pem /etc/openldap/cacerts/cacert.pem
chown ldap:ldap /etc/openldap/cacerts/cacert.pem
chmod 644 /etc/openldap/cacerts/cacert.pem
```

ldap 서버설정(slapd.conf)에 다음 내용을 추가한다. global 섹션에 추가하면 된다.

```
TLSCipherSuite HIGH:MEDIUM:+SSLv2  -> openssl ciphers
TLSCACertificateFile /etc/openldap/cacerts/cacert.pem -> CA private key
TLSCertificateFile /etc/openldap/slapdcert.pem -> certificate
TLSCertificateKeyFile /etc/openldap/slapdkey.pem -> private key
```

LDAP 서버에서 /etc/openldap/ldap.conf 에 아래 내용을 추가한다.

```
TLS_CACERTDIR /etc/openldap/cacerts
#TLS_REQCERT allow
```

TLS_REQCERT 는 TLS 세션에서 서버 인증서 체크와 연관된 부분이다. allow는 서버인증서가 없거나 잘못되어도 세션이 진행된다. TLS_REQCERT 에서 demand로 하면 서버인증서를 요청하되 서버인증서가 없거나 인증서가 잘못되었으면 세션을 바로 끊는다. (man ldap.conf)
ldap 서버를 내렸다가 다시 올려준다.

이제 ldap 클라이언트에서 다음의 설정을 /etc/ldap.conf에 한다. 여기서 cacert.pem은 ldap 클라이언트 시스템에 복사를 해두어야 한다.

```
ssl start_tls
tls_checkpeer yes
tls_cacertfile /etc/openldap/cacerts/cacert.pem
```

tls_checkpeer 서버 certificate 를 필요로 하고 검증을 하도록 한다. (설정파일의 주석내용 참고)

참고로 클라이언트 설정에서 authconfig를 이용하면

```
tls_cacertdir /etc/openldap/cacerts
```

로 설정이 된다. 위와 같이 tls_cacertfile 옵션을 이용하여 직접 파일을 지정할 수도 있고 아니면 /etc/opeanldap/cacerts 파일에 해당 인증서를 넣어두면 authconfig 에서 자동으로 c_rehah 유틸리티를 이용하여 해당 디렉토리에서 인증서파일을 가리키는 심볼릭 링크를 만든다.

```
# ls -alF /etc/openldap/cacerts
total 16
drwxr-xr-x  2 root root 4096 Jan  4 13:15 ./
drwxr-xr-x  4 root root 4096 Jan  4 13:18 ../
-rw-r--r--  1 root root 1346 Jan  4 13:15 cacert.pem
lrwxrwxrwx  1 root root   10 Jan  4 13:14 cc9fe289.0 -> cacert.pem
```

여기서 인증서파일을 가리키는 심볼릭 링크는 명령어를 이용할 수도 있다. 스크립트로 설정을 할 경우 필요하다.

```
/usr/sbin/cacertdir_rehash /etc/openldap/cacerts/
```

자신이 편한대로 쓰면 되겠지만 authconfig 를 이용한다면 자동으로 생성되는 tls_cacertdir 옵션을 써도 될 것이다.

## 클라이언트 설정시 주의점

클라이언트 설정은 앞에서 설명한 것과 동일하지만 ldap서버를 이용하여 접속할 각 클라이언트에서는 ldap 서버의 ca 인증서를 /etc/openldap/cacerts/ 디렉토리에 넣어두어야 한다.

그리고 바로 위에서 설명한대로 authconfig를 이용하여 설정하면 해당 인증서에 대한 심볼릭 링크를 자동으로 생성하지만 수동으로 할 경우에는 위의 cacertdir_rehash 스크립트를 이용한다.

# replication 구현

## 주의사항

openldap은 원래 single master replication system이다. 업데이트는 마스터에서만 되고 나머지는 읽기전용이라는 것이다.현재 openldap에서는 multimaster 를 지원하는 않는다.
replication에도 두가지 방식이 있으며 기존에 사용하던 slurpd와 최근부터 지원한 LDAP Sync Replication 이 있다. 현재는 slurpd만 테스팅을 하였다.
슬레이브에서 LDAP서버를 내리는 테스팅결과 잠시동안 네트워크등의 문제가 있다고 하더라도 슬레이브가 정상으로 돌아오면 리플리케이션이 정상적으로 동작되었다. 그렇지만 몇분이내의 간단한 테스팅만 한 것이므로 이것만을 가지고 신뢰성을 확인하기는 힘들 것이다. 그런데 네트워크의 이상등으로 연결이 되어있지 못할때 마스터에서 새로운 값을 입력하면 이는 나중에 연결이 복구되더라도 자동으로 슬레이브에 들어가지는 않는다.

## LDAP Sync Replication

LDAP Sync Replication 은 consumer-side replication으로 마스터서버(provider 서버)의 설정을 변경하거나 재시작하지 않고도 replicat를 생성할 수 있어 편리하다.
slurpd 방식에 비해 여러가지 장점이 있는 듯 하지만 RHEL이나 CentOS 4.4 에 기본 설치되어 있는 openldap 2.2 대에서는 몇가지 제약이 있어 실제로 쓰기는 불편한 듯 하다. 이 기능이 필요하다면 소스로 설치하여 해결할 수 있을 듯 한데 개인적으로는 이 기능이 당장 절실히 필요한 것은 아니라서 추가 테스팅은 하지 않았다. 2.2대와 2.3대에서 구현할때 약간의 차이점, 제약이 있다.

```
http://www.openldap.org/doc/admin22/syncrepl.html (openldap 2.2 ###)
While slapd (8) can function as the LDAP Sync provider only when it is configured with either
back-bdb or back-hdb backend, the syncrepl engine, which is a consumer-side replication engine,
can work with any backends.

http://www.openldap.org/doc/admin23/syncrepl.html (openldap 2.3###)
The syncrepl engine, which is a consumer-side replication engine, can work with any backends.
The LDAP Sync provider can be configured as an overlay on any backend, but works best with the
back-bdb or back-hdb backend. The provider can not support refreshAndPersist mode on back-ldbm
due to limits in that backend's locking architecture.
```

2.2 에서 마스터서버는 백엔드로 back-bdb, back-hdb 가 필요하고 슬레이브에서는 백엔드 제한이 없다. rpm 패키지에는 back-bdb 가 동작하지 않았으며 이에 대한 지원은 빠져있는 듯하다. 2.3 에서는 이러한 제한이 없다. 그렇지만 2.3에서도 백엔드로 back-bdb 나 back-hdb를 추천하고 있다.

설정하는 방법도 약간의 차이가 있으며 이는 매뉴얼을 참고한다.

## 구현순서

- 마스터서버의 slapd 대몬 내림
- 마스터서버의 slapd.conf 설정
- 마스터서버의 데이타를 슬레이브에 복사하고 슬레이브 서버에 넣어줌 (이경우 슬레이브 서버는 내려가 있다고 가정하고 이후에 세부 설정함)
- 슬레이브서버의 slpad.conf를 설정
- 슬레이브서버의 slapd 시작
- 마스터서버의 slapd 시작
- 마스터서버의 slurpd 시작 (centOS 에서는 replica 설정이 있는 경우 시작스크립트에서 자동으로 slapd, slurpd 함께 시작함)

## 마스터서버 설정

마스터서버에서는 아래의 내용을 /etc/openldap/sldapd.conf 에 추가한다.

```
replogfile /var/lib/ldap/openldap-master-replog
replica uri=ldap://cent.tunelinux.pe.kr:389
        suffix="dc=samjung,dc=com"
        binddn="cn=replica,dc=samjung,dc=com"
        credentials=xxxx
        bindmethod=simple
        tls=yes
```

replogfile 은 마스터서버에서 slapd가 로그 변화를 기록하는 파일이다. 이 파일을 slurpd가 읽어서 슬레이브 서버로 보낸다.
replica 를 이용하여 각 슬레이브 서버를 지정한다.

- uri : 슬레이브 서버 및 포트
- suffix : suffix
- binddn : 슬레이브 서버의 sldapd.conf 에서 updatedn 과 일치해야한다. 슬레이브 서버에서 이 권한을 가지고 마스터서버에서 오는 로그를 기록한다. 마스터서버의 rootdn과는 당연히 다르게 하는것이 좋을 것이다.
- bindmethod는 슬레이브와 통신을 하는데 사용하며 simple, sasl 을 선택할 수 있다. 여기서는 simple을 선택하였으며 credentials 는 슬레이브 서버에 바인드하기 위한 패스워드이다. 이는 슬레이브서버에서 지정한 것을 넣으면 된다.
- tls 는 마스터서버와 슬레이브서버간의 통신을 암호화한다.
- 마스터서버에서 데이타를 슬레이브서버로 옮기는 경우에 ldap서버를 내리고 slapcat 을 이용하여 LDIF 파일형태로 옮길 수 있다. 리플리카(슬레이브)에서는 slapadd 를 이용하여 데이타를 복원하면 된다. 그전에 slapd.conf 설정은 되어있어야 할 것이다.

```
root@master# slapcat -b "dc=samjung,dc=com" -l contents.ldif
... contents.ldif# ##### ####.
root@replica# slapadd -l contents.ldif
```

## 슬레이브서버 설정

```
> rootdn                "cn=replica,dc=samjung,dc=com"
> rootpw                {SSHA}IgT24XXXXEGN9aaLhBduKPJCp
> updatedn      "cn=replica,dc=samjung,dc=com"
> updateref     ldap://cent3.tunelinux.pe.kr
```

- updatedn : 마스터서버의 설정과 일치해야한다. updatedn은 해당 데이타에 쓰기 권한이 있어야 한다.
- updateref : 클라이언트에게 마스터 디렉토리 서버를 알려주는 URL. 클라이언트가 업데이트 요청을 하는 경우 마스터서버를 알려준다.

## 리플리케이션시 작동방식

클라이언트에서는 /etc/ldap.conf 의 host 에 master, slave 서버를 모두 지정해준다. 슬레이브에서는 updateref를 이용하여 슬레이브에 업데이트요청시 마스터서버로 업데이트 요청을 보낸다. 예를 들어 위에서 people에 속한 사용자의 경우 자신의 패스워드를 변경할 수가 있다. 이경우 slave 서버에서 자신의 패스워드를 변경할 경우 이에 대한 요청은 마스터로 가고 마스터에서 업데이트한후 다시 슬레이브서버로 동기화가 된다. 단, rootdn은 직접 작동하였다.

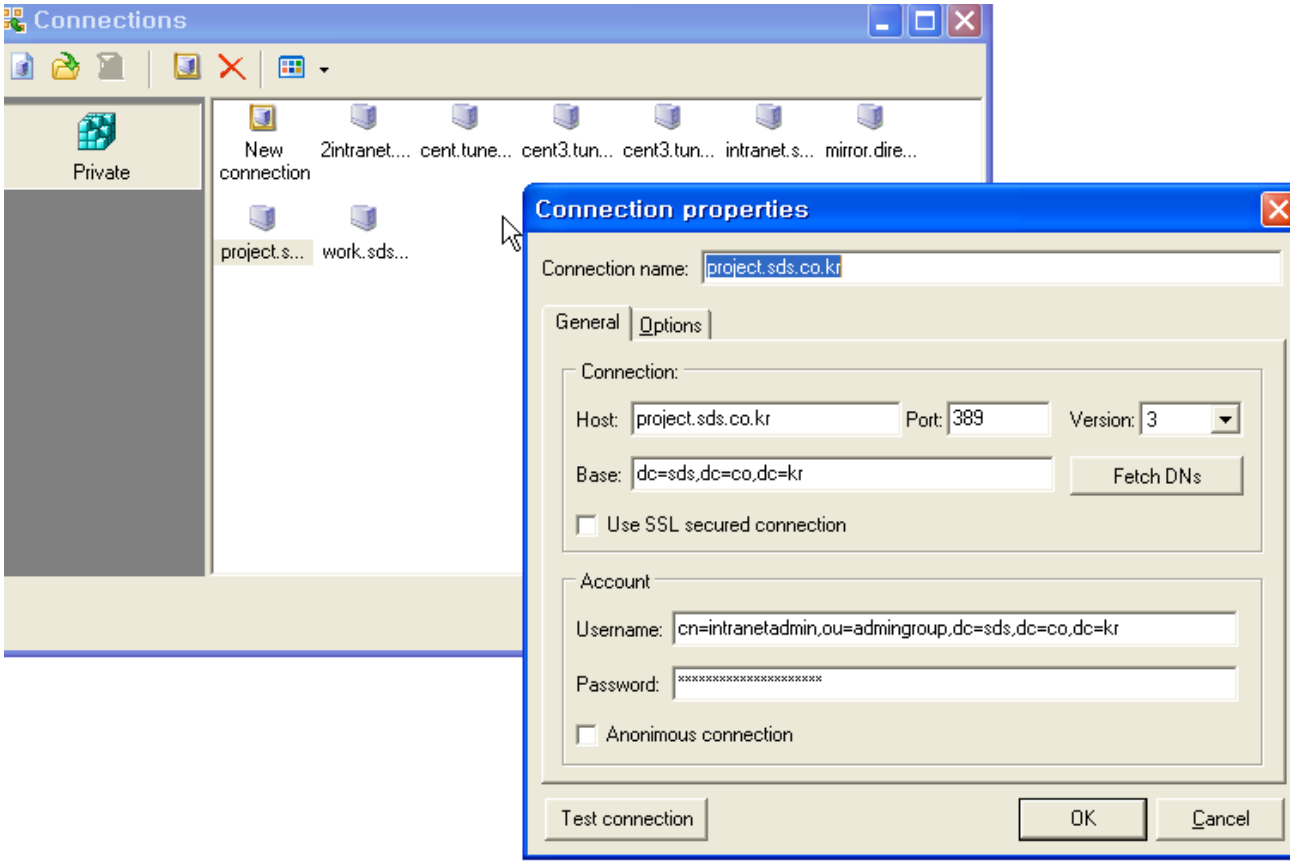
# 추가정보

## GUI tool

- http://ldapadmin.sourceforge.net/ ldap 검색, 수정 등 할 수 있는 윈도우 공개프로그램(GPL) 이 실제 사용해보니 편리함. GUI에서 사용자 이동, 복사, 그룹에 여러 사용자 추가등 가능함

아래 화면과 같이 설정을 한다.
Host : ldap 서버 주소
Port : 기본 389 (ssl 체크할 경우 자동 변경됨)
Base : Base dn . 여기 예제에서는 dc=samjung,dc=com

Username : 접속할 dn
Password : 비밀번호




- phpLDAPadmin (php), LDAP Account Manager(LAM, php), LDAP Browser(자바)등으로 된 프로그램이 있으나 사용하기에는 불편함
  LDAP Account Manager: lam.sourceforge.net
  phpLDAPadmin: http://sourceforge.net/projects/phpldapadmin/ . 소스포지에 가장 최신 버전이 올라가있음.
  LDAP Browser: www-unix.mcs.anl.gov/~gawor/ldap

## 로그확인

sldapd.conf 에서 loglevel 을 설정한다.
296 = 256 log connections/operations/results + 32 search filter processing + 8 connection management

```
loglevel        256
```

LDAP은 LOG_LOCAL4 facility를 사용하므로 /etc/syslog.conf 에 아래의 설정을 한다. ldap만 별도 파일로 저장할 수도 있다. 이 경우에는 로그로테이션을 주기적으로 해주어야 한다.

```
# grep local4 /etc/syslog.conf
local4.*                                                /var/log/messages
```

이경우 syslogd 를 다시 재시작해주어야 한다.

참고로 openldap 문서에도 로그레벨에 대한 내용은 있지만 남겨진 로그를 어떻게 분석하면 되는지에 대해서는 상세한 설명은 없었다. 이에 대해서는 작동방식은 비슷할 것이라 여겨지므로 레드햇 디렉토리 서버의 매뉴얼을 참고하면 될 듯하다. 이에 대한 내용은 http://www.redhat.com/docs/manuals/dir-server/ 레드햇 디렉토리 서버 매뉴얼 중에서 Configuration, Command, and File Reference 의 Chapter 5 Access Log and Connection Code Reference 를 참고한다. 여기서 로그에 남는 기록이 어떤 에러코드인지 설명을 참고하자.

## 동적인 서버설정 지원

openldap 2.3에서는 slapd.conf 설정도 LDIF 형태를 지원한다. 그래서 운영중인 상태에서도 ldap 서버의 설정값을 변경할 수 있다. 현재 CentOS 4.4 에 있는 rpm은 2.2 버전이다.

## Object Class Types

Object Class Types 은 Structural , Auxiliary, Abstract 세가지가 있다.
주의사항으로는 LDAP 디렉토리의 각 엔트에는 하나의 Structural object class만 있어야 한다. (오렐리 LDAP admin 20페이지)

## 접근제어

http://www.openldap.org/doc/admin23/slapdconfig.html#Access%20Control 참고
아래의 경우는 userPassword는 사용자만 변경하도록 하고 다른 사람은 볼 수 없도록 하며 mobile 등의 속성은 사용자가 변경가능하되 특정 dn에 속하는 사용자가 읽을 수 있도록 하였다.
특정 서브트리마다 접근권한을 제한하였고 group을 이용하여 linuxadmin 그룹에 속한 사용자만이 ou가 group, hosts, netgroup 등에 쓰기를 할 수 있도록 하였다.

> 주의사항
>
> 접근제어에서 중요한 것은 특정 조건에 대한 것을 앞으로 해두어야 한다는 것이다.
> 포괄적인 것을 앞에 두면 뒤에 세부적인 접근제어가 있어도 앞의 내용이 적용이 되어 실제 세부적인 접근제어를 하지 못하게 된다.

```
access to dn.subtree="dc=samjung,dc=com" attr=userPassword
        by self write
        by * auth

access to dn.subtree="dc=samjung,dc=com"
attrs=mobile,homePhone,samjunghomePostalCode,homePostalAddress
        by self write
        by dn.subtree="dc=samjung,dc=com" read
```

```
access to dn.subtree="ou=people,dc=samjung,dc=com"
        by dn.subtree="dc=samjung,dc=com" read
        by dn="cn=intranetadmin,ou=admingroup,dc=samjung,dc=com" write

access to dn.subtree="ou=role,dc=samjung,dc=com"
        by dn.subtree="dc=samjung,dc=com" read
        by dn="cn=intranetadmin,ou=admingroup,dc=samjung,dc=com" write

access to dn.subtree="ou=group,dc=samjung,dc=com"
        by
group/groupOfUniqueNames/uniqueMember="cn=linuxadmin,ou=admingroup,dc=samjung,dc=com" write
        by dn.subtree="dc=samjung,dc=com" read

access to dn.subtree="ou=hosts,dc=samjung,dc=com"
        by
group/groupOfUniqueNames/uniqueMember="cn=linuxadmin,ou=admingroup,dc=samjung,dc=com" write
        by dn.subtree="dc=samjung,dc=com" read

access to dn.subtree="ou=netgroup,dc=samjung,dc=com"
        by
group/groupOfUniqueNames/uniqueMember="cn=linuxadmin,ou=admingroup,dc=samjung,dc=com" write
        by dn.subtree="dc=samjung,dc=com" read

access to dn.subtree="ou=admingroup,dc=samjung,dc=com"
        by dn.subtree="dc=samjung,dc=com" read

access to *
        by * auth
```

## db 생성, 관리프로그램

slapadd : 오프라인에서 데이타 추가
slapindex : 오프라인에서 인덱스 재생성. slapd.conf 에서 설정이 바뀐 경우 기존 인덱스가 자동으로 변경되지 않는다. 이러한 경우 필요하다.
slapcat : 오프파인에서 데이타를 LDIF 형태로 덤프뜰때 사용. 백업시 편리함.

## nscd 네임서비스 캐싱 대몬 사용하기

nscd는 NIS, DNS 등의 네임서비스를 캐싱할 수 있는데 /etc/nscd.conf 에서 기본설정은 passwd, group, hosts 가 지정되어 있다. LDAP과 연동을 하는 경우 nscd를 사용하여 좀더 빠른 결과를 얻을 수 있을 것이다.

## inetOrgPersion 이용하여 개인정보 이용하기

Object Class에서 위는 account를 이용하였다. 그런데 accout는 다양한 개인정보를 담기에 는 불편하다. 예를 들어 email 등. 이럴 경우 account 대신 inetOrgPersion Object class를 이용할 수 있다. 여기서는 cn, sn이 필수요소이다. cpu 프로그램에서는 사용자의 오브젝트 클래스 지정하는 부분에서 account를 아래와 같이 inetOrgPerson 으로 바꾸어주고 사용자를 추가하거나 변경할 때 몇가지 옵션을 더 이용해야 한다. 위에서 말을 한대로 Structural Object는 한가지만 사용할 수 있는데 accout, inetOrgPerson은 Structural 오브젝트 클래스이고 posixAccount,shadowAccount,top 는 AUXILIARY 오브젝트 클래스이다.

```
#USER_OBJECT_CLASS      = account,posixAccount,shadowAccount,top
USER_OBJECT_CLASS       = inetOrgPerson,posixAccount,shadowAccount,top
```

-f (firstname,cn), -e (lastname, sn) -e (email) 옵션을 이용하면 된다.

```
# cpu useradd joo --firstname=# --lastname=## -e joon@sds.co.kr
User joo successfully added!
```

## 스키마 확장

스키마를 확장할 경우 attribute type 과 object classes 에 대한 유일한 OID를 할당하고 스키마 파일을 생성하여 slapd.conf에 포함시켜주면 된다. 내부적으로만 쓴다면 OID를 등록하지 않아도 되지만 외부와 함께 쓸 경우에는 충달할 수가 있다.

OID는 1.3.6.1.4.1. 이후가 private/enterprise 영역에 속한다. 아래와 같이 추가하고 이 파일을 slapd.conf에 추가해주면 여기서 만든 스키마를 사용할 수 있다. 아래는 참고예제로만 사용하길 바란다. plainjoePerson object class는 person을 상속받되 userPassword 와 mail은 필수로 있어야 하지만 새로 정의한 plainjoePath 는 없어도 괜찮다. 1.3.6.1.4.1.7777.1.1.2.1 은 OID, EQULITY 와 SUBSTR은 검색(matching rule)과 관련되어있으며 SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 는 dbms에서 데이타타입과 비슷하게 생각하면 된다.

```
attributetype ( 1.3.6.1.4.1.7777.1.1.2.1  NAME 'plainjoePath'
        DESC 'A directory on disk'
        SUBSTR caseExactIA5SubstringsMatch
        EQUALITY caseExactIA5Match
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 SINGLE-VALUE )

objectclass ( 1.3.6.1.4.1.7777.1.1.1.1 NAME 'plainjoePerson'
        SUP person STRUCTURAL
        MUST (userPassword $ mail)
        MAY ( plainjoePath ) )
```

openldap에서 스키마 확장에 대한 부분은 openldap 사이트의 http://www.openldap.org/doc/admin23/schema.html#Extending%20Schema 에서 확인하면 된다.

오렐리 ldap admin 책에서는 95쪽을 참고한다.

현재 삼정은 예전에 ldap 에 등록한 oid가 있으며 http://www.iana.org/assignments/enterprise-numbers 에서 확인이 가능하다. 등록한 oid는 1.3.6.1.4.1.7290이다.

http://www.openldap.org/doc/admin23/schema.html#Extending%20Schema 에서 제공하는 가이드라인에 따라 구성하면 다음과 같다.
1.3.6.1.4.1.7290 samjung
1.3.6.1.4.1.7290.1 SNMP
1.3.6.1.4.1.7290.2 LDAP
1.3.6.1.4.1.7290.2.1 attributes
1.3.6.1.4.1.7290.2.1.1 myattributes
1.3.6.1.4.1.7290.2.2 ObjectClasses
1.3.6.1.4.1.7290.2.2.1 myObjectClasses

# SSL 인증서 만들어서 여러 서비스에 활용하기

ldap에서 별도로 인증키를 만들어 사용하다가 apache 등에서 함께 이용하는 방법을 찾아보았다.
SSL 관련하여 다루어야 할 것이 여러가지 있기에 별도의 장으로 뺐다.

## 자체사인한 ssl 인증서 생성 스크립트

self 사인한 ssl 인증서를 생성하는 스크립트이다.

```
#!/bin/bash
# joon
```

```
# ## ### #### #### ##### 1825#(5#)## # ###.
# ####### root# ###### ldap #### ### ### ldap ### #### ######.
KEY_NAME='wiki.sds.co.kr'
KEY_DIRECTORY=/usr/local/direct/ssl
CA_DIRECTORY=$KEY_DIRECTORY/cacerts
PUBLIC_KEY=$CA_DIRECTORY/$KEY_NAME.crt
PRIVATE_KEY=$KEY_DIRECTORY/$KEY_NAME.key
DAYS="1825"

[ -f $PUBLIC_KEY ] && mv -fv $PUBLIC_KEY $PUBLIC_KEY.orig
[ -f $PRIVATE_KEY ] && mv -fv $PRIVATE_KEY $PRIVATE_KEY.orig

[ ! -d $KEY_DIRECTORY ] && mkdir -p $KEY_DIRECTORY
[ ! -d $CA_DIRECTORY ] && mkdir -p $CA_DIRECTORY

cd /etc/openldap
ln -s $CA_DIRECTORY .

SUBJ='/C=KR/ST=gurogu/L=seoul/O=samjung/OU=itservice/CN='$KEY_NAME

/usr/bin/openssl req -new -nodes -keyout $PRIVATE_KEY -x509 -days $DAYS -out $PUBLIC_KEY -subj
$SUBJ

/bin/chmod 644 $PUBLIC_KEY*
/bin/chown ldap.ldap $PRIVATE_KEY
/bin/chmod 400 $PRIVATE_KEY*

/etc/init.d/ldap restart
/etc/init.d/httpd restart
/usr/sbin/cacertdir_rehash $CA_DIRECTORY
```

위와 같은 작업을 한후 /etc/openldap/slapd.conf 에서 다음과 같이 설정한다.

```
TLSCipherSuite HIGH:MEDIUM:+SSLv2
TLSCACertificateFile /usr/local/direct/ssl/cacerts/wiki.sds.co.kr.crt
TLSCertificateFile /usr/local/direct/ssl/cacerts/wiki.sds.co.kr.crt
TLSCertificateKeyFile /usr/local/direct/ssl/wiki.sds.co.kr.key
```

웹서버설정에서는 아래과 같이 지정한다.
/etc/httpd/conf.d/ssl.conf

```
SSLCertificateFile /usr/local/direct/ssl/cacerts/wiki.sds.co.kr.crt
SSLCertificateKeyFile /usr/local/direct/ssl/wiki.sds.co.kr.key
```

## 키생성하기

아래는 개인키를 mirror.key 로 인증서를 mirror.crt로 만드는 경우이다.
nodes 옵션은 서버 재시작시 비밀번호를 입력하지 않도록 하기 위해서 사용한다.
root ca를 별도로 생성하지 않고 x509 옵션을 이용하여 자체 사인하여 사용하는 것이다.
mirror.crt 가 public key 이고 자체사인한 것이기 때문에 공인받지 않은 root ca 로 생각하면 될 듯 하다.
자체인증하여 사용하므로 days는 일부러 길게 주었다.

```
openssl req -new -nodes -keyout mirror.key -x509 -days 1825 -out mirror.crt
```

cn 도메인네임은 운영하려는 서버에 해당하는 것을 넣으면 된다.

## openldap 에서 이용하기

openldap에서는 위에서 만든 것을 아래와 같이 지정해준다.CA Certi 와 Certi 가 동일하다.
/etc/openldap/test 에 인증서를 놓아둔 경우이다.

```
TLSCACertificateFile /etc/openldap/test/mirror.crt
TLSCertificateFile /etc/openldap/test/mirror.crt
TLSCertificateKeyFile /etc/openldap/test/mirror.key
```

사용자 인증으로 ldap 을 사용할 경우 /etc/ldap.conf 에서 tls_cacertdir 이 /etc/openldap/cacerts 로 지정되어 있는 경우 mirror.crt 파일을 이 디렉토리로 복사해주고 cacertdir_rehash 명령을 실행해주어야 한다.

```
/usr/sbin/cacertdir_rehash /etc/openldap/cacerts/
```

# apache 에서 ssl 이용하기

## apache ssl 설정

apache 에서 ssl 설정은 다음과 같다. https로 테스팅을 해보기 바란다.
centos 에서 rpm으로 설치한 경우 /etc/httpd/conf.d/ssl.conf 파일을 조정한다.

```
SSLCertificateFile /etc/openldap/test/mirror.crt
SSLCertificateKeyFile /etc/openldap/test/mirror.key
```

여기에서 mirror.crt 파일은 공인인증을 받지 않았으므로 윈도우에서 crt 파일을 클릭한 경우 인증서가져오기(확장자를 .crt로 하면 됨)를 실행하여 인증서를 저장하면 된다.
그러면 웹브라우즈에서도 인증서 설치한 이후에는 편리하게 사용할 수 있다.

## apache 에서 ldap인증을 사용할 경우

apache 에서 ldap 인증을 사용하는 경우에는 아래 내용이 먼저 httpd.conf에 있어야한다. CA키로 mirror.crt를 지정하는 것이다.

LDAPTrustedCA /etc/openldap/test/mirror.crt
LDAPTrustedCAType BASE64_FILE

이제 htaccess 파일에서 아래와 같이 지정을 하면 된다.

```
# cat .htaccess
AuthType Basic
AuthName "Samjung Staff Only"
AuthLDAPBindDN cn=readonly,ou=admingroup,dc=samjung,dc=com
AuthLDAPBindPassword readonly
AuthLDAPURL ldaps://xxxx.direct.co.kr/ou=people,dc=samjung,dc=com?uid?sub?(objectClass=*)
require valid-user
```

# 주소록

아웃룩등의 주소록에서는 기타 설정에서 SSL사용을 체크해주면 된다.

## ldapadmin 프로그램

자체서명해서 사용하면 ldapadmin 에서는 셀프사인한 root 인증서라는 주의사항이 나온다. 이것은 불편해서 그냥 사용을 해야 할 듯하다.

## SSL/TLS 사용하여 php프로그래밍하기

- LDAP의모든것2 문서에 넣은 내용이지만 관련이 있기에 여기에도 넣는다.
- SSL/TLS에 대하여
  SSL과 TLS는 다른 부분입니다.
  오고가는 패킷을 암호화하는 것은 SSL/TLS 두가지 방법이 있습니다.

오렐리 LDAP 서적 26쪽에 나와있는 내용입니다.

- LDAP over SSL (LDAPS - 636 port) 일반적으로 말하는 SSL입니다.
- TLS : 389 PORT 로 암호화해서 통신하는 기능으로 RFC 2830 에서 추가된 기능이라고 합니다. 389 동일한 포트에서 암호화 되지 않은 패킷과 암호화된 패킷을 동시에 전송할 수 있는 기능입니다.

- SSL/TLS 사용시 php로 프로그래밍을 할 경우 주의사항
  php에는 ssl 을 지원하는 636 port 로 접속할 경우 host 에 ldaps 로 지정하고 포트를 636 지정합니다. start_tls는 명시하지 않습니다. 오히려 명시하면 에러가 나지요.

TLS를 이용하는 경우는 389 PORT 자체적으로 암호화를 지원하기 때문에 389 포트를 이용합니다. PHP 의 경우는 기본적으로 프로토콜 2를 사용하기 때문에 프로토콜 3을 사용하도록 옵션을 세팅하고 호스트에서는 ldaps 가 아닌 ldap으로 지정을 합니다. 반드시 start_tls 옵션을 사용해야지요.

ldaps 와 start_tls 기능은 별개입니다.

그리고 SSL이나 TLS를 이용하는 경우 접속하려는 호스트에서 서버의 인증서가 있어야 합니다. 리눅스의 경우 /etc/openldap/cacerts 디렉토리에 ca인증서가 있으며 LDAP서버의 CA인증서를 복사해놓고 authconfig 를 이용하여 ldap 사용을 다시 지정해주어야 합니다. 이에 대해서는 openldap 에서 SSL/TLS 부분을 참고합니다.

다시 정리하면 다음과 같습니다.

- PHP에서는 기본적으로 프로토콜 2를 사용하기 때문에 프로토콜 3을 사용하도록 옵션을 세팅
- SSL을 사용할 경우 636 포트를 사용하고 서버는 ldaps:// 으로 지정합니다.
- TLS를 사용할 경우 389 포트를 사용하고 서버는 ldap:// 으로 지정합니다. ldap_start_tls 를 지정해줘야 사용이 가능합니다.
- ldap_start_tls 는 ldap 서버에 연결(connect)한후 바인드(ldap_bind) 하기전에 지정합니다.
- 포트를 지정하지 않을 경우 SSL이냐 TLS냐에 따라 자동으로 포트를 선택합니다.

참고자료 : php 에서 start_tls 사용하기 http://kr.php.net/ldap_start_tls

- 참고코드
  위의 검색코드와 비교해보면 됩니다. 아래는 TLS를 이용할 경우이므로 만약 SSL 636 포트를 사용하고자 할 경우에는 서버를 ldaps로 지정하고 포트를 636으로 바꾸어준다. ldap_start_tls는 지정하면 안된다.

```
$target="cent3";

if($target=="aaaa")
{
        $ldaphost="ldap://aaaa.co.kr/";
        //$ldaphost="ldaps://aaaa.co.kr/";
        $ldap_port=389;
```

```
        //$ldap_port=636;
        $userdn="cn=manager,dc=co,dc=kr";
        $password="aaaaaa";
        $basedn="ou=people,dc=dc=co,dc=kr";
}

if($target=="cent3")
{
        //$ldaphost="ldaps://localhost/";
        $ldaphost="ldap://localhost/";
        //$ldap_port=636;
        $ldap_port=389;
        $userdn="cn=manager,dc=com";
        $password="bbbbb";
        $basedn="ou=people,dc=com";
}

if($ldap_port==="")
        $ldap_port="389";

// ####
$ds=ldap_connect($ldaphost,$ldap_port) or die("Could not connect to {$ldaphost}");  // must be
a valid LDAP server!
//$ds=ldap_connect($ldaphost) or die("Could not connect to {$ldaphost}");  // must be a valid
LDAP server!
if ($ds) {
   if (!ldap_set_option($ds, LDAP_OPT_PROTOCOL_VERSION, 3)) {
     fatal_error("Failed to set LDAP Protocol version to 3, TLS not supported.");
   }
}

if (!ldap_start_tls($ds)) {
    echo "failed start_tls\n";
    exit;
}

// id, password #### bind
$r=ldap_bind($ds,$userdn,$password) or die("Bind error");

// ### attr ##. ### ## ## ## attr# ## ###. ### ### ### attr# ###### ##
$justthese = array("uid","dn","cn", "mail",
"gidNumber","uidNumber","o","pager","shadowExpire","userpassword");
```

# 기타

## apache ldap인증의 경우 실제 서버세팅

AuthLDAPBindPassword 만 실제 비밀번호를 넣으면 된다.

```
AuthType Basic
AuthName "joon wiki system"
AuthLDAPURL ldap://hostname:389/ou=people,dc=sds,dc=co,dc=kr?uid?sub?(objectClass=*)
require valid-user
AuthLDAPBindDN cn=readonly,ou=admingroup,dc=sds,dc=co,dc=kr
AuthLDAPBindPassword xxxxx
```

## ldap 연동시 apache configure 옵션

```
--enable-ldap --enable-auth-ldap \
--with-ldap \
```