# Server OpenLDAP

LDAP è un acronimo per "Lightweight Directory Access Protocol", una versione semplificata del protocollo X.500. La directory impostata in questa sezione sarà usata per l'autenticazione. LDAP può comunque essere usato in diversi modi: autenticazione, directory condivisa (per i client mail), rubrica indirizzi. ecc...

Per descrivere LDAP velocemente, tutte le informazioni vengono archiviate in una struttura ad albero. Con **OpenLDAP** si ha la libertà di scegliere lo sviluppo dell'albero delle directory (il "Directory Information Tree", DIT). Per iniziare, si prende un esempio di un albero basilare con due nodi al di sotto della radice.

- Il nodo «People» è dove i propri utenti vengono salvati
- Il nodo «Groups» è dove i propri gruppi vengono salvati

Prima di iniziare, è necessario determinare quale sarà la radice della propria directory LDAP. In modo predefinito, l'albero sarà determinato dal proprio FQDN (Fully Qualified Domain Name), se il domino è "example.com" (usato in questo esempio), la radice sarà "dc=example,dc=com".

## Installazione

Per prima cosa, installare il demone server **OpenLDAP slapd** e **ldap-utils**, un pacchetto contenente le utilità di gestione LDAP:

```
sudo apt-get install slapd ldap-utils
```

By default **slapd** is configured with minimal options needed to run the **slapd** daemon.

The configuration example in the following sections will match the domain name of the server. For example, if the machine's Fully Qualified Domain Name (FQDN) is ldap.example.com, the default suffix will be *dc=example,dc=com*.

## Popolare LDAP

**OpenLDAP** uses a separate directory which contains the *cn=config* Directory Information Tree (DIT). The *cn=config* DIT is used to dynamically configure the **slapd** daemon, allowing the modification of schema definitions, indexes, ACLs, etc without stopping the service.

The backend *cn=config* directory has only a minimal configuration and will need additional configuration options in order to populate the frontend directory. The frontend will be populated with a "classical" scheme that will be compatible with address book applications and with Unix Posix accounts. Posix accounts will allow authentication to various applications, such as web applications, email Mail Transfer Agent (MTA) applications, etc.

> *Affinché le applicazioni esterne possano autenticarsi via LDAP, è necessario che siano configurate a tal fine. Per come fare, fare riferimento alla documentazione di ogni singola applicazione.*

> *Remember to change dc=example,dc=com in the following examples to match your LDAP configuration.*

First, some additional schema files need to be loaded. In a terminal enter:

```
sudo ldapadd -Y EXTERNAL -H ldapi:/// -f /etc/ldap/schema/cosine.ldif
sudo ldapadd -Y EXTERNAL -H ldapi:/// -f /etc/ldap/schema/nis.ldif
sudo ldapadd -Y EXTERNAL -H ldapi:/// -f /etc/ldap/schema/inetorgperson.ldif
```

Next, copy the following example LDIF file, naming it `backend.example.com.ldif`, somewhere on your system:

```
# Load dynamic backend modules
dn: cn=module,cn=config
objectClass: olcModuleList
cn: module
olcModulepath: /usr/lib/ldap
olcModuleload: back_hdb

# Database settings
dn: olcDatabase=hdb,cn=config
objectClass: olcDatabaseConfig
objectClass: olcHdbConfig
olcDatabase: {1}hdb
olcSuffix: dc=example,dc=com
olcDbDirectory: /var/lib/ldap
olcRootDN: cn=admin,dc=example,dc=com
olcRootPW: secret
olcDbConfig: set_cachesize 0 2097152 0
olcDbConfig: set_lk_max_objects 1500
olcDbConfig: set_lk_max_locks 1500
olcDbConfig: set_lk_max_lockers 1500
olcDbIndex: objectClass eq
olcLastMod: TRUE
olcDbCheckpoint: 512 30
olcAccess: to attrs=userPassword by dn="cn=admin,dc=example,dc=com" write by anonymous auth by self write by * none
olcAccess: to attrs=shadowLastChange by self write by * read
olcAccess: to dn.base="" by * read
```

```
olcAccess: to * by dn="cn=admin,dc=example,dc=com" write by * read
```

*Change olcRootPW: secret to a password of your choosing.*

Now add the LDIF to the directory:

```
sudo ldapadd -Y EXTERNAL -H ldapi:/// -f backend.example.com.ldif
```

The frontend directory is now ready to be populated. Create a `frontend.example.com.ldif` with the following contents:

```
# Create top-level object in domain
dn: dc=example,dc=com
objectClass: top
objectClass: dcObject
objectclass: organization
o: Example Organization
dc: Example
description: LDAP Example

# Admin user.
dn: cn=admin,dc=example,dc=com
objectClass: simpleSecurityObject
objectClass: organizationalRole
cn: admin
description: LDAP administrator
userPassword: secret

dn: ou=people,dc=example,dc=com
objectClass: organizationalUnit
ou: people

dn: ou=groups,dc=example,dc=com
objectClass: organizationalUnit
ou: groups

dn: uid=john,ou=people,dc=example,dc=com
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: shadowAccount
uid: john
sn: Doe
givenName: John
cn: John Doe
displayName: John Doe
uidNumber: 1000
gidNumber: 10000
userPassword: password
gecos: John Doe
loginShell: /bin/bash
homeDirectory: /home/john
shadowExpire: -1
shadowFlag: 0
shadowWarning: 7
shadowMin: 8
shadowMax: 999999
shadowLastChange: 10877
mail: john.doe@example.com
postalCode: 31000
l: Toulouse
o: Example
mobile: +33 (0)6 xx xx xx xx
homePhone: +33 (0)5 xx xx xx xx
title: System Administrator
postalAddress:
initials: JD

dn: cn=example,ou=groups,dc=example,dc=com
objectClass: posixGroup
cn: example
gidNumber: 10000
```

In questo esempio sono stati impostati la struttura della directory, un utente e un gruppo. In altri esempi potrebbe essere possibile notare, in ogni voce, l'elemento *objectClass: top*, ma dato che è il comportamento predefinito, non è necessario inserirlo esplicitamente.

Add the entries to the LDAP directory:

```
sudo ldapadd -x -D cn=admin,dc=example,dc=com -W -f frontend.example.com.ldif
```

We can check that the content has been correctly added with the **ldapsearch** utility. Execute a search of the LDAP directory:

```
ldapsearch -xLLL -b "dc=example,dc=com" uid=john sn givenName cn

dn: uid=john,ou=people,dc=example,dc=com
cn: John Doe
sn: Doe
givenName: John
```

Una semplice spiegazione:

- *-x*: non usa il metodo di autenticazione, predefinito, SASL.
- *-LLL*: disabilita la stampa di informazioni sullo schema LDIF.

## Further Configuration

L'albero *cn=config* può essere manipolato usando le utilità presenti nel pacchetto **ldap-utils**. Per esempio:

- Usare **ldapsearch** per visualizzare l'albero, inserendo la password dell'amministratore impostata durante l'installazione o la riconfigurazione:

```
sudo ldapsearch -LLL -Y EXTERNAL -H ldapi:/// -b cn=config dn
```

```
SASL/EXTERNAL authentication started
SASL username: gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth
SASL SSF: 0
dn: cn=config

dn: cn=module{0},cn=config

dn: cn=schema,cn=config

dn: cn={0}core,cn=schema,cn=config

dn: cn={1}cosine,cn=schema,cn=config

dn: cn={2}nis,cn=schema,cn=config

dn: cn={3}inetorgperson,cn=schema,cn=config

dn: olcDatabase={-1}frontend,cn=config

dn: olcDatabase={0}config,cn=config

dn: olcDatabase={1}hdb,cn=config
```

The output above is the current configuration options for the *cn=config* backend database. Your output may be vary.

- Come esempio per modificare un albero *cn=config*, aggiungere un altro attributo all'indice usando il comando **ldapmodify**:

```
sudo ldapmodify -Y EXTERNAL -H ldapi:///
```

```
SASL/EXTERNAL authentication started
SASL username: gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth
SASL SSF: 0
dn: olcDatabase={1}hdb,cn=config
add: olcDbIndex
olcDbIndex: uidNumber eq

modifying entry "olcDatabase={1}hdb,cn=config"
```

Una volta completata la modifica, premere *Ctrl+D* per uscire dall'utilità:

- **ldapmodify** è anche in grado di leggere le modifica da un file. Copiare e incolalre quanto segue in un file chiamato `uid_index.ldif`:

```
dn: olcDatabase={1}hdb,cn=config
add: olcDbIndex
olcDbIndex: uid eq,pres,sub
```

Eseguire **ldapmodify**:

```
sudo ldapmodify -Y EXTERNAL -H ldapi:/// -f uid_index.ldif
```

```
SASL/EXTERNAL authentication started
SASL username: gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth
SASL SSF: 0
modifying entry "olcDatabase={1}hdb,cn=config"
```

Qusto metodo è molto utile per applicare grandi modifiche.

- Adding additional *schemas* to **slapd** requires the schema to be converted to LDIF format. The `/etc/ldap/schema` directory contains some schema files already converted to LDIF format as demonstrated in the previous section. Fortunately, the **slapd** program can be used to automate the conversion. The following example will add the *dyngoup.schema*:

  1. Creare un file di conversione `schema_convert.conf` contenente le seguenti righe:

```
include /etc/ldap/schema/core.schema
include /etc/ldap/schema/collective.schema
include /etc/ldap/schema/corba.schema
include /etc/ldap/schema/cosine.schema
include /etc/ldap/schema/duaconf.schema
include /etc/ldap/schema/dyngroup.schema
include /etc/ldap/schema/inetorgperson.schema
include /etc/ldap/schema/java.schema
include /etc/ldap/schema/misc.schema
include /etc/ldap/schema/nis.schema
include /etc/ldap/schema/openldap.schema
include /etc/ldap/schema/ppolicy.schema
```

  2. Creare una directory temporanea in cui salvare l'output:

```
mkdir /tmp/ldif_output
```

3. Utilizzando **slapcat**. convertire il file schema in LDIF:

```
slapcat -f schema_convert.conf -F /tmp/ldif_output -n0 -s "cn={5}dyngroup,cn=schema,cn=config" >
/tmp/cn=dyngroup.ldif
```

Nel caso i nomi della directory temporanea e del file siano diversi, modificarli di conseguenza. Potrebbe essere comunque utile mantenere la directory `ldif_output` per aggiungere altri schema in un secondo momento.

4. Edit the `/tmp/cn\=dyngroup.ldif` file, changing the following attributes:

```
dn: cn=dyngroup,cn=schema,cn=config
...
cn: dyngroup
```

Rimuovere le seguenti righe dalla fine del file:

```
structuralObjectClass: olcSchemaConfig
entryUUID: 10dae0ea-0760-102d-80d3-f9366b7f7757
creatorsName: cn=config
createTimestamp: 20080826021140Z
entryCSN: 20080826021140.791425Z#000000#000#000000
modifiersName: cn=config
modifyTimestamp: 20080826021140Z
```


*I valori degli attributi possono variare, basta solo assicurarsi che gli attributi siano rimossi.*

5. In fine, usando l'utilità **ldapadd**, aggiugnere il nuovo schema alla directory:

```
sudo ldapadd -Y EXTERNAL -H ldapi:/// -f /tmp/cn\=dyngroup.ldif
```

There should now be a *dn: cn={4}dyngroup,cn=schema,cn=config* entry in the cn=config tree.

## LDAP Replication

LDAP diventa spesso un servizio altamente critico all'interno della rete e diversi sistemi possono dipendere su LDAP per autenticazione, autorizzazioni, configurazione, ecc... In questi casi è molto utile impostare un sistema ridondante attraverso l'uso della replicazione.

Replication is achieved using the *Syncrepl* engine. Syncrepl allows the changes to be synced using a *consumer*, *provider* model. A provider sends directory changes to consumers.

### Provider Configuration

The following is an example of a *Single-Master* configuration. In this configuration one OpenLDAP server is configured as a *provider* and another as a *consumer*.

1. First, configure the provider server. Copy the following to a file named `provider_sync.ldif`:

```
# Add indexes to the frontend db.
dn: olcDatabase={1}hdb,cn=config
changetype: modify
add: olcDbIndex
olcDbIndex: entryCSN eq
-
add: olcDbIndex
olcDbIndex: entryUUID eq

#Load the syncprov and accesslog modules.
dn: cn=module{0},cn=config
changetype: modify
add: olcModuleLoad
olcModuleLoad: syncprov
-
add: olcModuleLoad
olcModuleLoad: accesslog

# Accesslog database definitions
dn: olcDatabase={2}hdb,cn=config
objectClass: olcDatabaseConfig
objectClass: olcHdbConfig
olcDatabase: {2}hdb
olcDbDirectory: /var/lib/ldap/accesslog
olcSuffix: cn=accesslog
olcRootDN: cn=admin,dc=example,dc=com
olcDbIndex: default eq
olcDbIndex: entryCSN,objectClass,reqEnd,reqResult,reqStart

# Accesslog db syncprov.
dn: olcOverlay=syncprov,olcDatabase={2}hdb,cn=config
changetype: add
objectClass: olcOverlayConfig
objectClass: olcSyncProvConfig
olcOverlay: syncprov
olcSpNoPresent: TRUE
olcSpReloadHint: TRUE

# syncrepl Provider for primary db
dn: olcOverlay=syncprov,olcDatabase={1}hdb,cn=config
changetype: add
```

```
objectClass: olcOverlayConfig
objectClass: olcSyncProvConfig
olcOverlay: syncprov
olcSpNoPresent: TRUE

# accesslog overlay definitions for primary db
dn: olcOverlay=accesslog,olcDatabase={1}hdb,cn=config
objectClass: olcOverlayConfig
objectClass: olcAccessLogConfig
olcOverlay: accesslog
olcAccessLogDB: cn=accesslog
olcAccessLogOps: writes
olcAccessLogSuccess: TRUE
# scan the accesslog DB every day, and purge entries older than 7 days
olcAccessLogPurge: 07+00:00 01+00:00
```

2. The **AppArmor** profile for **slapd** will need to be adjusted for the accesslog database location. Edit `/etc/apparmor.d/usr.sbin.slapd` adding:

```
  /var/lib/ldap/accesslog/ r,
  /var/lib/ldap/accesslog/** rwk,
```

Then create the directory, reload the **apparmor** profile, and copy the `DB_CONFIG` file:

```
sudo -u openldap mkdir /var/lib/ldap/accesslog
sudo -u openldap cp /var/lib/ldap/DB_CONFIG /var/lib/ldap/accesslog/
sudo /etc/init.d/apparmor reload
```

*Using the -u openldap option with the **sudo** commands above removes the need to adjust permissions for the new directory later.*

3. Edit the file and change the *olcRootDN* to match your directory:

```
olcRootDN: cn=admin,dc=example,dc=com
```

4. Next, add the LDIF file using the **ldapadd** utility:

```
sudo ldapadd -Y EXTERNAL -H ldapi:/// -f provider_sync.ldif
```

5. Restart **slapd**:

```
sudo /etc/init.d/slapd restart
```

The *Provider* server is now configured, and it is time to configure a *Consumer* server.

### Consumer Configuration

1. On the *Consumer* server configure it the same as the *Provider* except for the *Syncrepl* configuration steps.

Add the additional schema files:

```
sudo ldapadd -Y EXTERNAL -H ldapi:/// -f /etc/ldap/schema/cosine.ldif
sudo ldapadd -Y EXTERNAL -H ldapi:/// -f /etc/ldap/schema/nis.ldif
sudo ldapadd -Y EXTERNAL -H ldapi:/// -f /etc/ldap/schema/inetorgperson.ldif
```

Also, create, or copy from the provider server, the `backend.example.com.ldif`

```
# Load dynamic backend modules
dn: cn=module,cn=config
objectClass: olcModuleList
cn: module
olcModulepath: /usr/lib/ldap
olcModuleload: back_hdb

# Database settings
dn: olcDatabase=hdb,cn=config
objectClass: olcDatabaseConfig
objectClass: olcHdbConfig
olcDatabase: {1}hdb
olcSuffix: dc=example,dc=com
olcDbDirectory: /var/lib/ldap
olcRootDN: cn=admin,dc=example,dc=com
olcRootPW: secret
olcDbConfig: set_cachesize 0 2097152 0
olcDbConfig: set_lk_max_objects 1500
olcDbConfig: set_lk_max_locks 1500
olcDbConfig: set_lk_max_lockers 1500
olcDbIndex: objectClass eq
olcLastMod: TRUE
olcDbCheckpoint: 512 30
olcAccess: to attrs=userPassword by dn="cn=admin,dc=example,dc=com" write by anonymous auth by self write by *
none
olcAccess: to attrs=shadowLastChange by self write by * read
olcAccess: to dn.base="" by * read
olcAccess: to * by dn="cn=admin,dc=example,dc=com" write by * read
```

And add the LDIF by entering:

```
sudo ldapadd -Y EXTERNAL -H ldapi:/// -f backend.example.com.ldif
```

2. Do the same with the `frontend.example.com.ldif` file listed above, and add it:

```
sudo ldapadd -x -D cn=admin,dc=example,dc=com -W -f frontend.example.com.ldif
```

The two severs should now have the same configuration except for the *Syncrepl* options.

3. Now create a file named `consumer_sync.ldif` containing:

```
#Load the syncprov module.
dn: cn=module{0},cn=config
changetype: modify
add: olcModuleLoad
olcModuleLoad: syncprov

# syncrepl specific indices
dn: olcDatabase={1}hdb,cn=config
changetype: modify
add: olcDbIndex
olcDbIndex: entryUUID eq
-
add: olcSyncRepl
olcSyncRepl: rid=0 provider=ldap://ldap01.example.com bindmethod=simple binddn="cn=admin,dc=example,dc=com"
 credentials=secret searchbase="dc=example,dc=com" logbase="cn=accesslog"
 logfilter="(&(objectClass=auditWriteObject)(reqResult=0))" schemachecking=on
 type=refreshAndPersist retry="60 +" syncdata=accesslog
-
add: olcUpdateRef
olcUpdateRef: ldap://ldap01.example.com
```

You will probably want to change the following attributes:

- *ldap01.example.com* to your server's hostname.
- *binddn*
- *credentials*
- *searchbase*
- *olcUpdateRef:*

4. Add the LDIF file to the configuration tree:

```
sudo ldapadd -c -Y EXTERNAL -H ldapi:/// -f consumer_sync.ldif
```

The frontend database should now sync between servers. You can add additional servers using the steps above as the need arises.

*Il demone **slapd** invia, in modo predefinito, informazioni di registro a `/var/log/syslog`. Se qualche cosa non dovesse funzionare controllare quel file per eventuali errori e informazioni su come poterli risolvere. Inoltre, assicurarsi che tutti i server possano raggiungere quel FQDN (Fully Qualified Domain Name). Questo viene configurato nel file `/etc/hosts` in questo modo:*

```
127.0.0.1       ldap01.example.com ldap01
```

.

## Impostare ACL

L'autenticazione richiede accesso al campo della password che non dovrebbe essere accessibile in modo predefinito. Inoltre, affinché gli utenti possa cambiare la loro password usando passwd o altre utilità, *shadowLastChange* deve essere accessibile una volta che l'utente si è autenticato.

Per visualizzare l'ACL (Access Control List), usare l'utilità **ldapsearch**:

```
ldapsearch -xLLL -b cn=config -D cn=admin,cn=config -W olcDatabase=hdb olcAccess
```

```
Enter LDAP Password:
dn: olcDatabase={1}hdb,cn=config
olcAccess: {0}to attrs=userPassword,shadowLastChange by dn="cn=admin,dc=exampl
 e,dc=com" write by anonymous auth by self write by * none
olcAccess: {1}to dn.base="" by * read
olcAccess: {2}to * by dn="cn=admin,dc=example,dc=com" write by * read
```

## TLS e SSL

Durante la fase di autenticazione a un server OpenLDAP, è raccomandato usare una sessione cifrata. Questo può essere ottenuto usando TLS (Transport Layer Security) o SSL (Secure Sockets Layer).

The first step in the process is to obtain or create a *certificate*. Because **slapd** is compiled using the **gnutls** library, the **certtool** utility will be used to create certificates.

1. First, install **gnutls-bin** by entering the following in a terminal:

```
sudo apt-get install gnutls-bin
```

2. Next, create a private key for the *Certificate Authority* (CA):

```
sudo sh -c "certtool --generate-privkey > /etc/ssl/private/cakey.pem"
```

3. Create a `/etc/ssl/ca.info` details file to self-sign the CA certificate containing:

```
cn = Example Company
ca
cert_signing_key
```

4. Now create the self-signed CA certificate:

```
sudo certtool --generate-self-signed --load-privkey /etc/ssl/private/cakey.pem \
 --template  /etc/ssl/ca.info --outfile /etc/ssl/certs/cacert.pem
```

5. Make a private key for the server:

```
sudo sh -c "certtool --generate-privkey > /etc/ssl/private/ldap01_slapd_key.pem"
```

> *Replace ldap01 in the filename with your server's hostname. Naming the certificate and key for the host and service that will be using them will help keep filenames and paths straight.*

6. To sign the server's certificate with the CA, create the `/etc/ssl/ldap01.info` info file containing:

```
organization = Example Company
cn = ldap01.example.com
tls_www_server
encryption_key
signing_key
```

7. Create the server's certificate:

```
sudo certtool --generate-certificate --load-privkey /etc/ssl/private/x01-test_slapd_key.pem \
 --load-ca-certificate /etc/ssl/certs/cacert.pem --load-ca-privkey /etc/ssl/private/cakey.pem \
 --template /etc/ssl/x01-test.info --outfile /etc/ssl/certs/x01-test_slapd_cert.pem
```

Una volta ottenuto un certificato, la chiave e si è installato il tutto, usare **ldapmodify** per aggiungere le nuove opzioni di configurazione:

```
sudo ldapmodify -Y EXTERNAL -H ldapi:///
```

```
Enter LDAP Password:
dn: cn=config
add: olcTLSCACertificateFile
olcTLSCACertificateFile: /etc/ssl/certs/cacert.pem
-
add: olcTLSCertificateFile
olcTLSCertificateFile: /etc/ssl/certs/ldap01_slapd_cert.pem
-
add: olcTLSCertificateKeyFile
olcTLSCertificateKeyFile: /etc/ssl/private/ldap01_slapd_key.pem

modifying entry "cn=config"
```

> *Adjust the `ldap01_slapd_cert.pem`, `ldap01_slapd_key.pem`, and `cacert.pem` names if yours are different.*

Aprire il file `/etc/default/slapd` e togliere il commento dall'opzione *SLAPD_SERVICES*:

```
SLAPD_SERVICES="ldap:/// ldapi:/// ldaps:///"
```

Garantire accesso al certificato all'utente *openldap*:

```
sudo adduser openldap ssl-cert
sudo chgrp ssl-cert /etc/ssl/private/ldap01_slapd_key.pem
sudo chmod g+r /etc/ssl/private/ldap01_slapd_key.pem
```

> *Se la directory `/etc/ssl/private` e il file `/etc/ssl/private/server.key` hanno permessi diversi, modificare i comandi secondo le proprie esigenze.*

In fine, riavviare **slapd**:

```
sudo /etc/init.d/slapd restart
```

Il demone **slapd** dovrebbe ora essere in ascolto per le connessioni LDAPS e dovrebbe essere in grado di usare STARTTLS nella fase di autenticazione.

> *Se il server non dovesse avviarsi, controllare in /var/log/syslog. Se sono presenti degli errori come «main: TLS init def ctx failed: -1», potrebbe esserci un problema di configurazione. Controllare che il certificato sia firmato dalla stessa autorità dei file configurati e che il gruppo ssl-cert abbia permessi di lettura sulla chiave privata.*

### Replicazione TLS

Se è stato impostato **Syncrepl** tra i server è utile cifrare il traffico di replicazione usando *TLS* (Transport Layer Security). Per maggiori informazioni su come impostare la replicazione, consultare la sezione chiamata «LDAP Replication».

Assuming you have followed the above instructions and created a CA certificate and server certificate on the *Provider* server. Follow the following instructions to create a certificate and key for the *Consumer* server.

1. Create a new key for the Consumer server:

```
mkdir ldap02-ssl
cd ldap02-ssl
certtool --generate-privkey > ldap02_slapd_key.pem
```

*Creating a new directory is not strictly necessary, but it will help keep things organized and make it easier to copy the files to the Consumer server.*

2. Next, create an info file, `ldap02.info` for the Consumer server, changing the attributes to match your locality and server:

```
country = US
state = North Carolina
locality = Winston-Salem
organization = Example Company
cn = ldap02.salem.edu
tls_www_client
encryption_key
signing_key
```

3. Create the certificate:

```
sudo certtool --generate-certificate --load-privkey ldap02_slapd_key.pem \
 --load-ca-certificate /etc/ssl/certs/cacert.pem --load-ca-privkey /etc/ssl/private/cakey.pem \
 --template ldap02.info --outfile ldap02_slapd_cert.pem
```

4. Copy the `cacert.pem` to the dicretory:

```
cp /etc/ssl/certs/cacert.pem .
```

5. The only thing left is to copy the `ldap02-ssl` directory to the Consumer server, then copy `ldap02_slapd_cert.pem` and `cacert.pem` to `/etc/ssl/certs`, and copy `ldap02_slapd_key.pem` to `/etc/ssl/private`.

6. Once the files are in place adjust the *cn=config* tree by entering:

```
sudo ldapmodify -Y EXTERNAL -H ldapi:///
```

```
Enter LDAP Password:
dn: cn=config
add: olcTLSCACertificateFile
olcTLSCACertificateFile: /etc/ssl/certs/cacert.pem
-
add: olcTLSCertificateFile
olcTLSCertificateFile: /etc/ssl/certs/ldap02_slapd_cert.pem
-
add: olcTLSCertificateKeyFile
olcTLSCertificateKeyFile: /etc/ssl/private/ldap02_slapd_key.pem

modifying entry "cn=config"
```

7. As with the Provider you can now edit `/etc/default/slapd` and add the *ldaps:///* parameter to the *SLAPD_SERVICES* option.

Now that *TLS* has been setup on each server, once again modify the *Consumer* server's *cn=config* tree by entering the following in a terminal:

```
sudo ldapmodify -Y EXTERNAL -H ldapi:///
```

```
SASL/EXTERNAL authentication started
SASL username: gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth
SASL SSF: 0

dn: olcDatabase={1}hdb,cn=config
replace: olcSyncrepl
olcSyncrepl: {0}rid=0 provider=ldap://ldap01.example.com bindmethod=simple binddn="cn=ad
 min,dc=example,dc=com" credentials=secret searchbase="dc=example,dc=com" logbas
 e="cn=accesslog" logfilter="(&(objectClass=auditWriteObject)(reqResult=0))" s
 chemachecking=on type=refreshAndPersist retry="60 +" syncdata=accesslog starttls=yes

modifying entry "olcDatabase={1}hdb,cn=config"
```

Se il nome host del server LDAP non corrisponde al FQDN (Fully Qualified Domain Name) nel certificato, potrebbe essere necessario modificare il file `/etc/ldap/ldap.conf` e aggiungere le seguenti opzioni TLS:

```
TLS_CERT /etc/ssl/certs/ldap02_slapd_cert.pem
TLS_KEY /etc/ssl/private/ldap02_slapd_key.pem
TLS_CACERT /etc/ssl/certs/cacert.pem
```

Riavviare **slapd** su ogni server:

```
sudo /etc/init.d/slapd restart
```

## Autenticazione LDAP

Una volta ottenuto un server LDAP funzionante, i pacchetti **auth-client-config** e **libnss-ldap** consentono di configurare facilmente un client Ubuntu affinché utilizzi l'autenticazione LDAP. Per installare questi pacchetti, digitare:

```
sudo apt-get install libnss-ldap
```

Durante la fase di installazione un dialogo richiederà i dettagli relativi alla connessione al server LDAP.

Nel caso si commetta un errore durante l'inserimento delle informazioni, è possibile rieseguire la configurazione tramite il comando:

```
sudo dpkg-reconfigure ldap-auth-config
```

I risultati della configurazione possono essere visualizzati nel file `/etc/ldap.conf`. Se il server richiede delle opzioni non contemplate durante la fase di configurazione. modificare il file secondo le proprie esigenze.

Ora che **libnss-ldap** è configurato, abilitare il profilo **auth-client-config** LDAP digitando:

```
sudo auth-client-config -t nss -p lac_ldap
```

- *-t*: modifica solamente `/etc/nsswitch.conf`.
- *-p*: nome del profilo da abilitare, disabilitare, ecc...
- *lac_ldap*: il profilo **auth-client-config** parte del pacchetto **ldap-auth-config**.

Utilizzando l'utilità **pam-auth-update**, configurare il sistema affinché utilizzi LDAP per l'autenticazione:

```
sudo pam-auth-update
```

Dal menù **pam-auth-update**, scegliere LDAP e qualsiasi altro metodo di autenticazione necessario.

Ora dovrebbe essere possibile eseguire l'accesso utilizzando le credenziali presenti nella directory LDAP.

*Se LDAP viene utilizzato per archiviare utenti Samba, è necessario configurare il server affinché utilizzi l'autenticazione via LDAP. Per maggiori informazioni, consultare la sezione chiamata «Samba e LDAP».*

## Gestire utenti e gruppi

Il pacchetto **ldap-utils** contiene diverse utilità per la gestione di directory, ma le molte opzioni necessarie possono rendere queste utilità di difficile utilizzo. Il pacchetto **ldapscripts** contiene degli script configurabili per gestire facilmente utenti e gruppi LDAP.

Per installare il pacchetto, da un terminale:

```
sudo apt-get install ldapscripts
```

Aprire il file `/etc/ldapscripts/ldapscripts.conf` e togliere il commento o aggiungere quanto segue in base alle proprie esigenze:

```
SERVER=localhost
BINDDN='cn=admin,dc=example,dc=com'
BINDPWDFILE="/etc/ldapscripts/ldapscripts.passwd"
SUFFIX='dc=example,dc=com'
GSUFFIX='ou=Groups'
USUFFIX='ou=People'
MSUFFIX='ou=Computers'
GIDSTART=10000
UIDSTART=10000
MIDSTART=10000
```

Creare il file `ldapscripts.passwd` per consentire l'accesso autenticato alla directory:

```
sudo sh -c "echo -n 'secret' > /etc/ldapscripts/ldapscripts.passwd"
sudo chmod 400 /etc/ldapscripts/ldapscripts.passwd
```

*Sostituire «secret» con la password dell'amministratore LDAP.*

Le applicazioni in **ldapscripts** sono ora pronte per la gestione delle directory. Quelli che seguono sono degli esempi sull'utilizzo di questi script:

- Creare un nuovo utente:

  ```
  sudo ldapadduser mario example
  ```

  Viene creato un utente con UID mario e imposta il gruppo primario (GID) dell'utente a example

- Cambiare la password di un utente:

  ```
  sudo ldapsetpasswd mario
  Changing password for user uid=mario,ou=People,dc=example,dc=com
  New Password:
  New Password (verify):
  ```

- Eliminare un utente:

  ```
  sudo ldapdeleteuser mario
  ```

- Aggiungere un gruppo:

  ```
  sudo ldapaddgroup qa
  ```

- Eliminare un gruppo:

  ```
  sudo ldapdeletegroup qa
  ```

- Aggiungere un utente a un gruppo:

```
sudo ldapaddusertogroup george qa
```

Dovrebbe essere possibile visualizzare un attributo *memberUid* per il gruppo qa con un valore di mario.

- Rimuovere un utente da un gruppo:

```
sudo ldapdeleteuserfromgroup george qa
```

L'attributo *memberUid* dovrebbe ora essere rimosso dal gruppo qa.

- Lo script **ldapmodifyuser** consente di aggiungere, rimuovere o replicare gli attributi di un utente. Lo script utilizza la stessa sintassi dell'utilità **ldapmodify**. Per esempio:

```
sudo ldapmodifyuser george
# About to modify the following entry :
dn: uid=george,ou=People,dc=example,dc=com
objectClass: account
objectClass: posixAccount
cn: george
uid: george
uidNumber: 1001
gidNumber: 1001
homeDirectory: /home/george
loginShell: /bin/bash
gecos: george
description: User account
userPassword:: e1NTSEF9eXFsTFcyWlhwWkF1eGUybVdFWHZKRzJVMjFTSG9vcHk=

# Enter your modifications here, end with CTRL-D.
dn: uid=george,ou=People,dc=example,dc=com
replace: gecos
gecos: Mario Rossi
```

L'utente *gecos* dovrebbe ora essere «Mario Rossi».

- Un'altra utile caratteristica di **ldapscripts** è il sistema dei modelli. I modelli consentono di personalizzare gli attributi di un utente, gruppo e degli oggetti macchina. Per esempio, per abilitare il modello *user* aprire il file `/etc/ldapscripts/ldapscripts.conf` modificando:

```
UTEMPLATE="/etc/ldapscripts/ldapadduser.template"
```

Diversi esempi sono disponibili nella directory `/etc/ldapscripts`. Copiare o rinominare il file `ldapadduser.template.sample` in `/etc/ldapscripts/ldapadduser.template`:

```
sudo cp /etc/ldapscripts/ldapadduser.template.sample /etc/ldapscripts/ldapadduser.template
```

Modificare il modello per aggiungere gli attributi desiderati. In questo esempio viene creato un nuovo utente con una *objectClass* di *inetOrgPerson*:

```
dn: uid=<user>,<usuffix>,<suffix>
objectClass: inetOrgPerson
objectClass: posixAccount
cn: <user>
sn: <ask>
uid: <user>
uidNumber: <uid>
gidNumber: <gid>
homeDirectory: <home>
loginShell: <shell>
gecos: <user>
description: User account
title: Employee
```

Notare l'opzione *<ask>* utilizzata per il valore *cn*. Usando <ask> viene configurato **ldapadduser** per chiedere il valore dell'attributo durante la creazione dell'utente.

Sono presenti molti altri script nel pacchetto. Per un elenco completo usare il comando: dpkg -L ldapscripts | grep bin

## Risorse

- The OpenLDAP Ubuntu Wiki page has more details.
- Per maggiori informazioni, consultare il sito web di OpenLDAP
- Anche se un po' datata, un'ottima fonte di informazioni riguardo LDAP è LDAP System Administration di O'Reilly.
- Il libro di Packt, Mastering OpenLDAP, è un'ottima fonte che copre anche le nuove versioni di OpenLDAP.
- Per maggiori informazioni su **auth-client-config**, consultare la pagina di manuale: man auth-client-config.
- Per maggiori informazioni riguardo il pacchetto **ldapscripts**, consultare le pagine di manuale: man ldapscripts, man ldapadduser, man ldapaddgroup, ecc...

Capitolo 6. Autenticazione di rete Samba e LDAP