LDAP Linux HOWTO

Luiz Ernesto Pinheiro Malère

malere@yahoo.com

LDAP Linux HOWTO

Luiz Ernesto Pinheiro Malère

Pubblicato v1.09, 5 marzo 2004

In questo documento sono presenti informazioni su come installare, configurare ed utilizzare un Server LDAP (Lightweight Directory Access Protocol) su una macchina Linux. Il documento inoltre presenta dettagli su come creare database LDAP, aggiungere, aggiornare e cancellare informazioni nella directory. Questo HOWTO è in gran parte basato su informazioni riprese dalla documentazione LDAP dell'università del Michigan e dalla Guida dell'Amministatore OpenLDAP. Traduzione e aggiornamenti a cura di Giulio Daprelà *giulio@pluto.it*. Revisione a cura di Hugh Hartmann *hhartmann@libero.it*

Diario delle Revisioni

Revisione 1.09 2004/03/05
OpenLDAP 2.2 and general corrections.
Revisione 1.08 2003/04/02
SASL with DIGEST-MD5 authentication.
Revisione 1.07 2002/09/16
Typo correction.
Revisione 1.06 2002/07/17
Migration to DocBook XML standard, revision of the role document. Introducing OpenLDAP 2.1.
Revisione 1.05 2001/06/22 Revisionato da: lepm
Correction of long lines that were causing inconsistences on the PDF version of the document.
Revisione 1.04 2001/02/28 Revisionato da: lepm
Correction of more typos and update on the following sections: Roaming Access, Authentication using LDAP.
Revisione 1.03 2000/09/28 Revisionato da: lepm
Presenting OpenLDAP 2.0, which comprises LDAPv3, as defined on RFC2251 (ftp://ftp.isi.edu/in-notes/rfc2251.txt)
Revisione 1.02 2000/09/13 Revisionato da: lepm
Correction of typos and addition of the section History of Releases.
Revisione 1.01 2000/02/15 Revisionato da: lepm
Added the following sections: LDAP Migration Tools, Authentication using LDAP, Graphical LDAP tools, RFCs.
Revisione 1.00 1999/06/20 Revisionato da: lepm
Initial version.

Sommario

Lista delle Tabelle

Capitolo 1. Introduzione

Lo scopo principale di questo documento è installare ed utilizzare un server di directory LDAP sulla propria macchina Linux. Si imparerà come installare, configurare, far funzionare e manutenere un server LDAP. Dopo di che si potrà anche imparare come poter immagazzinare, richiamare e aggiornare le informazioni sul proprio server usando i client e i programmi di utilità di LDAP. Il demone per il directory server LDAP è denominato *slapd* e funziona su molte piattaforme UNIX differenti.

C'è un altro demone che si occupa della replica fra i server LDAP. Esso è denominato slurpd e per il momento non ci si deve preoccupare del suo funzionamento. In questo documento si fa funzionare uno slapd che fornisce il servizio directory soltanto per il proprio dominio locale, senza replica, quindi senza slurpd. Le informazioni complete sulla replica sono disponibili nella: Guida dell'Amministatore OpenLDAP (http://www.openldap.org/doc/admin21/replication.html)

La messa a punto del dominio locale rappresenta una scelta semplice per la configurazione del proprio server, utile per cominciare, ma con cui è facile passare ad un'altra configurazione, se lo si desidera. Le informazioni presentate in questo documento rappresentano un buon inizio per l'uso del server LDAP. Probabilmente, dopo aver letto questo documento ci si sentirà incoraggiati ad espandere le capacità del proprio server e perfino scrivere proprie applicazioni client, usando i kit di sviluppo già disponibili per C, C++ e Java.

1.1. Che cosa è LDAP?

LDAP sta per Lightweight Directory Access Protocol. Come il nome suggerisce, è un protocollo leggero client-server per l'accesso ai servizi di directory, in particolare ai servizi di directory basati sullo standard X-500. LDAP funziona sopra il protocollo TCP/IP o su altri protocolli di rete orientati alla connessione. LDAP è definito nel documento RFC2251 (ftp://ftp.isi.edu/in-notes/rfc2251.txt) "The Lightweight Directory Access Protocol (v3).

Un servizio di directory è simile a un database, ma tende a contenere le informazioni sugli attributi in maniera più descrittiva. L'informazione in una directory è letta generalmente molto più spesso di quanto sia scritta. I servizi di directory sono ottimizzati per dare una rapida risposta ad accessi in grande quantità o per operazioni di ricerca. Possono avere la capacità di replicare ampiamente le informazioni, al fine di aumentare la disponibilità e l'affidabilità, mentre riducono il tempo di risposta. Quando l'informazione della directory è replicata, le inconsistenze momentanee fra le repliche possono essere tollerate, finchè non vengono finalmente sincronizzate.

Ci sono molti modi differenti di gestire un servizio di directory. Diverse metodologie permettono a generi differenti di informazioni di essere immagazzinate nelle directory, pongono differenti requisiti su come queste informazioni possano essere consultate, interrogate ed aggiornate, su come sono protette da accessi non autorizzati, ecc. Alcuni servizi di directory sono locali, forniscono cioè il servizio ad un contesto limitato (per esempio, il servizio finger su una singola macchina). Altri servizi di directory sono globali, e forniscono il servizio ad un contesto molto più ampio.

1.2. Come lavora LDAP?

Il servizio di directory LDAP è basato su un modello client-server. Uno o più server LDAP contengono i dati che costituiscono l'albero di directory di LDAP o il database LDAP sottostante. Un client LDAP si collega ad un server LDAP e fa una domanda. Il server o risponde, o indica al client dove poter ottenere maggiori informazioni (tipicamente, un altro server LDAP). Non importa a quale server LDAP un client si connette poiché esso avrà sempre la stessa vista dell'albero di directory; lo stesso nome su server LDAP diversi identifica un unico oggetto. Ciò è una caratteristica molto importante di un servizio di directory globale, come appunto LDAP.

1.3. LDAP backend, oggetti e attributi

Il demone del server LDAP è chiamato *Slapd*. *Slapd* supporta una varietà di differenti **database backends** utilizzabili.

Essi includono come scelta primaria **BDB**, un database backend transazionale ad alte prestazioni; LDBM, un backend leggero basato su DBM; SHELL, un'interfaccia di backend per script di shell arbitrari e PASSWD, una semplice interfaccia di backend per il file passwd(5).

BDB utilizza Sleepycat (http://www.sleepycat.com/) Berkeley DB 4. LDBM utilizza Berkeley DB (http://www.sleepycat.com/) o GDBM (http://www.gnu.org/software/gdbm/).

Il backend transazionale BDB è adatto per l'accesso a database di lettura/scrittura in modalità multiutente, con qualunque mix di operazioni lettura/scrittura. BDB è usato nelle applicazioni che richiedono:

- Transazioni, compresi cambiamenti multipli di database effettuati in modo atomico e il rollback di eventuali modifiche non completate, quando necessario.
- Capacità di ripristino da crash del sistema e da guasti hardware senza perdere nessuna transazione completata.

In questo documento si suppone che venga scelto il database BDB.

Per importare ed esportare le informazioni tra directory server basati su LDAP, o per descrivere un insieme di cambiamenti da apportare alla directory, si usa tipicamente il formato file LDIF (LDAP Data Interchange Format). Un file LDIF memorizza informazioni in strutture gerarchiche orientate agli oggetti. Il pacchetto di programmi di LDAP che si avrà è fornito di un programma di utilità per convertire i files LDIF in formato BDB.

Un file LDIF comune è simile a questo:

```
dn: o=TUDelft, c=NL
o: TUDelft
objectclass: organization
dn: cn=Luiz Malere, o=TUDelft, c=NL
cn: Luiz Malere
sn: Malere
mail: malere@yahoo.com
objectclass: person
```

Come si può vedere ogni voce è identificata unicamente in base a un nome distinto, o DN. Il DN è composto dal nome della voce più un percorso di nomi che risalgono dalla voce all'indietro fino in cima alla struttura gerarchica della directory (come in un albero).

In LDAP, una **classe oggetto** definisce la collezione degli **attributi** che possono essere usati per definire una voce. Lo standard di LDAP fornisce questi tipi base di classi oggetto:

- Gruppi nella directory, che includono liste non ordinante di singoli oggetti o gruppi di oggetti.
- Locazioni, come il nome di un paese e una descrizione.
- Organizzazioni nella directory.
- Persone nella directory.

Una voce può appartenere a più di una classe oggetto. Per esempio, la voce per una persona è definita dalla classe oggetto *person*, ma può anche essere definita dagli attributi delle classi inetOrgPerson, groupOfNames e organization. La struttura di classi oggetto del server (il suo schema) determina la lista totale degli attributi obbligatori e permessi per ogni singola voce.

I dati della directory sono rappresentati come coppie di attributi-valori. Qualsiasi informazione specifica è associata a un attributo descrittivo.

Per esempio, l'attributo commonName, o cn, è usato per immagazzinare il nome della persona. Una persona chiamata Jonas Salk può essere rappresentata nella directory come

```
cn: Jonas Salk
```

Ogni persona che è inserita nella directory è definita da una serie di attributi nella classe *person*. Altri attributi usati per definire questa voce potrebbero essere:

```
givenname: Jonas
surname: Salk
mail: jonass@airius.com
```

Gli attributi obbligatori includono gli attributi che devono essere presenti nella voce in quanto appartenenti ad una specifica classe. Tutte le voci richiedono l'attributo objcetClass, che elenca le classi cui l'oggetto appartiene.

Gli attributi consentiti includono gli attributi che possono essere presenti nelle voci che usano una classe oggetto. Per esempio, nella classe oggetto person, gli attributi cn e sn sono obbligatori. Gli attributi descrizione, telephoneNumber, seeAlso e userpassword sono consentiti ma non sono obbligatori.

Ogni attributo ha una corrispondente definizione di sintassi. La definizione di sintassi descrive il tipo di informazione fornita dall'attributo, per esempio:

- bin: tipo binario.
- ces: case exact string (durante il confronto le maiuscole/minuscole devono corrispondere).
- cis: case ignore string (durante il confronto maiuscole/minuscole sono ignorate).
- tel: telephone number string (come cis ma gli spazi in bianco ed il ' - ' sono ignorati durante i confronti).
- dn: distinguished name.

Nota: di solito le definizioni delle classi oggetto e degli attributi risiedono nei file di schema, nella sottodirectory *schema* sotto la directory di installazione di OpenLDAP.

1.4. Nuove versioni di questo documento

Questo documento può subire delle correzioni e aggiornamenti basati su feedback ricevuti dai lettori. Si può visitare l'url:

http://www.tldp.org/HOWTO/LDAP-HOWTO.html

per avere nuove versioni di questo HOWTO.

1.5. Opinioni e Suggerimenti

Se si ha qualunque genere di dubbio circa le informazioni presenti in questo documento, ci si metta in contatto con me al seguente indirizzo di email: malere@yahoo.com

Se avete commenti e/o suggerimenti, fatemi sapere!

1.6. Ringraziamenti

Questo Howto è il risultato di una internship fatta da me all'interno dell'università di TUDelft - Paesi Bassi. desidero ringraziare le persone che mi hanno incoraggiato a redigere questo documento: *Rene van Leuken* e *Wim Tiwon*. Grazie molto. Inoltre sono fan di Linux, proprio come me.

Desidero ringraziare anche Thomas Bendler, autore del Ldap-Howto tedesco, per il suo contributo al mio documento, e anche Joshua Go, grande volontario sul progetto LDP.

Karl Lattimer merita un premio, per il suo grande contributo sulle questioni relative a SASL.

E grazie a Dio!

1.7. Copyright e Disclaimer

Se avete domande, visitate il seguente url: http://www.gnu.org/licenses/fdl.txt e mettetevi in contatto con il coordinatore di Linux HOWTO, a: guylhem@metalab.unc.edu

Capitolo 2. Installazione del server LDAP

Sono necessarie cinque fasi per installare il server LDAP:

- Installare i pacchetti pre-requisiti (se non già installati).
- Scaricare l'applicazione server.
- Spacchettare il software.
- Configurare i Makefile.
- Compilare il server.

2.1. Pre-Requisiti

Per essere pienamente conformi alle specifiche di LDAPv3, i client e i server OpenLDAP richiedono l'installazione di qualche pacchetto addizionale. Per scrivere questo documento, ho usato una distribuzione Mandrake 9.0 con un Kernel 2.4.20, installando manualmente il pacchetto Berkeley BDB e le librerie SASL.

Librerie SASL OpenSSL TLS

Le librerie OpenSSL TLS sono normalmente parte del sistema base o costituiscono un componente software opzionale. l'url ufficiale OpenSSL TLS è: http://www.openssl.org

Servizi di Autenticazione Kerberos

I client e i server OpenLDAP supportano i servizi di autenticazione basati su Kerberos. In particolare OpenLDAP supporta il meccanismo di autenticazione SASL/GSSAPI usando i pacchetti Heimdal o MIT Kerberos V. Se si desidera usare l'autenticazione SASL/GSSAPI basata su kerberos, bisogna installare o heimdal o MIT kerberos V. Heimdal kerberos è disponibile all'indirizzo http://www.pdc.kth.se/heimdal MIT Kerberos è disponibile all'indirizzo http://web.mit.edu/kerberos/www

L'uso di servizi di autenticazione forti, simili a quelli forniti da Kerberos, è fortemente raccomandato.

Cyrus's Simple Authentication e Security Layer Libraries

Le librerie SASL del progetto Cyrus fanno parte normalmente del sistema di base o costituiscono un componente software opzionale. Cyrus SASL è disponibile all'indirizzo http://asg.web.cmu.edu/sasl/sasl-library.html. Cyrus SASL userà le librerie Kerberos/GSSAPI e OpenSSL se preinstallate. Al momento in cui sto scrivendo, ho usato Cyrus SASL 2.1.12.

Database Software

Il principale database backend di slapd, BDB, richiede la versione 4 dello Sleepycat Software Berkeley DB (http://www.sleepycat.com). Se non sarà disponibile al momento della configurazione, non potrete compilare lo slapd con il database backend principale.

Il vostro sistema operativo può contenere il DB di Berkeley, versione 4, nel sistema di base o come componente software opzionale. Altrimenti ci sono parecchie versioni disponibili su Sleepycat (http://www.sleepycat.com/download.html). Al momento in cui sto scrivendo, è consigliato l'utilizzo dell'ultima versione 4.1.25. Il backend LDBM del server slapd di OpenLDAP supporta una varietà di database manager, come Berkeley DB (versione 3) e GDBM. GDBM è disponibile dal sito di download della FSF ftp://ftp.gnu.org/pub/gnu/gdbm/ (ftp://ftp.gnu.org/pub/gnu/gdbm/.).

Thread

Il supporto thread è quasi certamente parte integrante del proprio sistema linux di base. OpenLDAP è progettato per trarre vantaggi dai thread. OpenLDAP supporta i POSIX pthread, i Mach Cthread, e un certo numero di altre varietà. Lo script *configure* protesterà se non troverà un sottosistema thread adatto. Se questo dovesse accadere, siete pregati di consultare la sezione software - Installation - Platform Hints dell'OpenLDAP FAQ: http://www.openldap.org/faq/.

TCP Wrapper

Slapd supporta TCP wrapper (filtri di controllo di accesso a livello IP) se preinstallati. L'uso dei TCP wrapper o di altri filtri di accesso a livello IP (come quelli forniti da una firewall a livello IP) è raccomandato per i server che contengono informazioni non pubbliche.

2.2. Scaricare il pacchetto

Ci sono due distribuzioni libere di server LDAP: LDAP server dell'Università del Michigan e OpenLDAP server. C'è anche il Netscape directory server, che è libero solo sotto qualche condizione (gli istituti scolastici possono averlo liberamente per esempio). OpenLDAP server è basato sull'ultima versione del Server dell'Università del Michigan e ci sono mailing list e documentazione supplementare disponibile per esso. Questo documento assume che si stia usando il server OpenLDAP.

L'ultima versione tar.gzip è disponibile al seguente indirizzo:

http://www.openldap.org

Se si vuole l'ultima versione del server dell'Università del Michigan, si vada al seguente indirizzo:

ftp://terminator.rs.itd.umich.edu/ldap

Per scrivere questo documento, ho usato la versione 2.1.16 del pacchetto OpenLDAP. Il mio sistema operativo è un Mandrake Linux 9.0 con kernel 2.4.20.

Sul sito di OpenLDAP potete sempre trovare le ultime versioni di sviluppo e stabili del server Open LDAP. Quando questo documento è stato aggiornato, l'ultima versione stabile era openldap-stable-20030317.tgz (versione 2.1.16). l'ultima versione sviluppata era invece openldap-2.1.16.tgz.

2.3. Spacchettare il Software

Ora che si ha il pacchetto tar.gzip sulla propria macchina locale, si può spacchettarlo.

Prima copiare il pacchetto in una directory di proprio gradimento, per esempio /usr/local. Poi usare il comando seguente:

```
tar xvzf openldap-2.1.16.tgz
```

Si può usare anche questo comando, equivalente:

```
gunzip openldap-2.1.16.tgz | tar xvf -
```

2.4. Configurare il Software

I sorgenti del server OpenLDAP sono distribuiti con uno script di configurazione per impostare opzioni quali le directory di installazione, i flag del compilatore e del linker. Eseguire il seguente comando nella directory in cui si è spacchettato il software:

```
./configure --help
```

Tutte le opzioni che si potranno personalezzare si otterranno con lo script di configurazione prima di copiare il software. Alcune opzioni utili sono: -- prefix=pref, -- exec-prefix=eprefix e -- bindir=dir, per impostare le directory di installazione. Normalmente se si lancia lo script di configurazione senza opzioni, esso rileverà le impostazioni appropriate e preparerà l'installazione nei percorsi di default. Quindi digitare semplicemente:

```
./configure
```

E guardare l'output per controllare se va tutto bene

Suggerimento: a volte bisogna passare delle opzioni specifiche al proprio script di configurazione, come per esempio -- with-tls (per permettere allo slapd di utilizzare un canale sicuro: LDAPS://). In questo caso, potrebbe darsi che le proprie librerie SSL/TLS risiedano in una directory non standard del proprio sistema. Si può informare lo script di configurazione dell'ubicazione delle librerie cambiando le proprie variabili d'ambiente, utilizzando il comando *env*. Esempio: si supponga di aver installato il pacchetto openssl sotto /usr/local/openssl. Il seguente comando compilerà slapd con il supporto SSL/TLS:

```
env CPPFLAGS=-I/usr/local/openssl/include \
      LDFLAGS=-L/usr/local/openssl/lib \
      configure --with-tls ...
```

si possono specificare le seguenti variabili d'ambiente con il comando *env* prima dello script di configurazione:

- CC: specifica un compilatore C alternativo.
- CFLAGS: specifica i flag addizionali per il compilatore.
- CPPFLAGS: specifica i flag per il Preprocessore C.
- LDFLAGS: specifica i flag per il linker.
- LIBS: specifica delle librerie addizionali.

2.5. Compilazione del Server

Dopo aver configurato il software si può iniziare a compilarlo. Prima compilare le dipendenze, usando il comando:

```
make depend
```

Dopo compilare il server, usando il comando:

```
make
```

Se tutto va bene, il server sarà compilato come da configurazione. Altrimenti, ritornare al punto precedente per rivedere le impostazioni di configurazione. Bisognerebbe leggere i file di README e di INSTALL situati nella directory in cui si è spacchettato il software. Inoltre controllare i suggerimenti specifici dello script di configurazione, situati nel percorso doc/install/configure sotto la directory dove si è spacchettato il software.

Per assicurare una corretta compilazione bisognerebbe lanciare la procedura di test (impiegherà alcuni minuti):

```
make test
```

I test che si applicano alla propria configurazione verranno eseguiti e dovrebbero essere superati. Alcune prove, quale la prova della replica, possono essere saltate.

Ora installare i file binari e le man page. Si può avere bisogno di essere super-user per fare questo (a seconda di dove si stanno installando le cose):

```
su root -c 'make install'
```

Questo è tutto; ora si ha il file binario del server e i file binari di parecchi altri programmi di utilità. Andare al Capitolo 3 per vedere come configurare il funzionamento del proprio server LDAP.

Capitolo 3. Configurazione del server LDAP

Una volta che il software è stato installato e compilato, si è pronti a configurarlo per l'uso che se ne deve fare. Tutta la configurazione runtime dello slapd è compiuta attraverso il file *slapd.conf*, installato nella directory prefix che si è specificato nello script di configurazione o di default in /usr/local/etc/openldap.

Questa sezione specifica in dettaglio le istruzioni di configurazione comunemente usate per *slapd.conf*. Per una lista completa, vedere la pagina di manuale di slapd.conf(5). Le istruzioni del file di configurazione sono divise in tre tipologie: **global**, **backend specific** e **database specific**. Qui si troveranno le descrizioni delle istruzioni, insieme ai loro valori di default (se ci sono) ed agli esempi di uso.

3.1. Formato del file di configurazione

Il file slapd.conf consta di tre tipi di informazioni di configurazione: global, backend specific, e database specific. Le informazioni globali sono specificate per prime, seguite dalle informazioni associate con un particolare tipo di backend, a loro volta seguite dalle informazioni associate ad una particolare istanza di database.

Le istruzioni globali possono essere sovrascritte da istruzioni di backend e/o di database, e le istruzioni di backend possono essere sovrascritte dalle istruzioni di database.

Le linee in bianco e le linee di commento che cominciano con il carattere '#' sono ignorate. Se una linea comincia con uno spazio viene considerata continuazione della riga precedente. Il formato generale di slapd.conf è il seguente:

```
# direttive globali di configurazione
<global config directives>

# definizione backend
backend <typeA>
<backend-specific directives>

# definizione primo database & direttive di configurazione
database <typeA>
<database-specific directives>

# definizione secondo database & direttive di configurazione
database <typeB>
<database-specific directives>

# definizione secondo database "typeA" & direttive di configurazione
database <typeA>
<database-specific directives>
```

```
# backend successivo & definizioni database & direttive di configurazione
...
```

Un'istruzione di configurazione può avere argomenti. In questo caso sono separati da spazio bianco. Se un argomento contiene uno spazio bianco, deve essere racchiuso tra doppi apici "come questo". Se una discussione contiene un doppio apice o il carattere di backslash '\', il carattere dovrebbe essere preceduto da un altro backslash '\'.

La distribuzione contiene un file di configurazione di esempio che sarà installato nella directory /usr/local/etc/openldap. Un certo numero di file che contengono le definizioni dello schema (tipi di attributo e classe oggetto) sono forniti nella directory /usr/local/etc/openldap/schema.

3.2. Istruzioni globali

Le istruzioni descritte in questa sezione si applicano a tutti i backends e database a meno che non siano specificamente sovrascritte in una definizione di backend o di database. Gli argomenti che dovrebbero essere sostituiti dal testo effettivo sono indicati tra parentesi angolari < >.

```
access to <what> [ by <who> <accesslevel> <control> ]+
```

Questa istruzione garantisce l'accesso (specificato da <accesslevel>) ad un insieme di voci e/o attributi (specificati da <what>) per uno o più richiedenti (specificati da <who>). Vedere gli esempi della la Sezione 3.7 per maggiori dettagli.

Importante: se non è specificata alcuna direttiva di accesso, la politica di controllo accesso di default, accesso * in * lettura, consente accesso in lettura sia agli utenti autenticati che a quelli anonimi.

```
attributetype <RFC2252 Attribute Type Description>
```

Questa direttiva definisce un tipo di attributo. Controllare il seguente URL per maggiori dettagli: Schema Specification (http://www.openldap.org/doc/admin/schema.html)

```
idletimeout <integer>
```

Specifica il numero di secondi di attesa prima di chiudere forzatamente una connessione client non utilizzata. Un idletimeout pari a 0, il default, disabilita questa caratteristica.

```
include <filename>
```

Questa istruzione implica che lo slapd legga le informazioni supplementari di configurazione dal file in questione prima di continuare con la linea successiva del file corrente. Il file incluso dovrebbe seguire le normali disposizioni del file config dello slapd. Il file è usato comunemente per includere file che contengono le specifiche dello schema.

Nota: si dovrebbe fare attenzione quando si usa questa istruzione - non c'è un limite basso al numero di direttive include annidate e non viene fatto nessun controllo di loop.

```
loglevel <integer>
```

Questa direttiva specifica il livello al quale dovrebbero essere inviate al file syslog le informazioni statistiche e i messaggi di debug (attualmente i messaggi vengono inviati al syslogd(8) con il servizio LOCAL4). Bisogna aver configurato OpenLDAP con l';opzione --enable-debug (di default) affinché questo funzioni (tranne per i due livelli di statistica, che sono sempre permessi). I livelli di log sono cumulativi. Per visualizzare quali numeri corrispondono a quale tipo di debug, richiamare lo slapd con -? o consultare la tabella sottostante. I valori possibili per <integer> sono:

Tabella 3-1. Livelli di debugging

Livello	Descrizione
-1	attiva tutti i livelli di debug
0	nessun debug
1	traccia le chiamate a funzione
2	debug del trattamento del pacchetto
4	tracciamento pesante
8	gestione della connessione
16	scrittura dei pacchetti spediti e ricevuti
32	eleborazione dei filtri di ricerca
64	elaborazione del file di configurazione
128	elaborazione dela lista di controllo d'accesso
256	statistiche delle connessioni-operazioni-risultati
512	statistiche degli oggetti inviati
1024	scrittura della comunicazione con la shell di backend
2048	scrittura di debug del parsing degli oggetti

Esempio:

loglevel 255 or loglevel -1

Questo farà sì che vengano loggate moltissime informazioni di debug.

Default:

loglevel 256

```
objectclass <RFC2252 Object Class Description>
```

Questa informazione definisce una classe oggetto. Controllare il seguente URL per maggiori informazioni: Schema Specification (http://www.openldap.org/doc/admin/schema.html)

```
referral <URI>
```

Questa istruzione specifica il riferimento da restituire quando lo slapd non può trovare un database locale per trattare una richiesta.

Esempio:

referral ldap://root.openldap.org

Questa istruzione rinvierà interrogazioni non-locali alla root globale del server LDAP del progetto OpenLDAP. Client LDAP intelligenti possono riformulare la richiesta a quel server, ma notare che la maggior parte di questi client stanno solo cominciando a capire come gestire semplici URL LDAP che contengono una parte con il nome dell'host e facoltativamente una parte con il Distiguished Name.

```
sizelimit <integer>
```

Questa istruzione specifica il numero massimo di oggetti che si otterranno dall'operazione di ricerca.

Default:

sizelimit 500

```
timelimit <integer>
```

Questa istruzione specifica il numero massimo di secondi (in tempo reale) che slapd impiegherà per rispondere alla richiesta di ricerca. Se una richiesta non è conclusa in questo lasso di tempo, il risultato sarà una segnalazione di superamento del tempo limite.

Default:

timelimit 3600

3.3. Direttive generali di backend

Le direttive contenute in questa sezione si applicano soltanto al backend nel quale sono definite. Sono sostenute da ogni tipo di backend. le istruzioni di backend si applicano a tutte le istanze di database dello stesso tipo e, secondo l'istruzione, possono essere sovrascritte dalle istruzioni del database.

```
backend <type>
```

Questa istruzione contrassegna l'inizio di una definizione backend. <type> dovrebbe essere o bdb o uno degli altri tipi di backend supportati ed elencati di seguito:

Tabella 3-2. Database Backend

Tipo	Descrizione
bdb	Berkeley DB transactional backend
dnssrv	DNS SRV backend
ldbm	Lightweight DBM backend
ldap	Lightweight Directory Access Protocol (Proxy) backend
meta	Meta Directory backend
monitor	Monitor backend
passwd	Fornisce l'accesso in sola lettura a passwd(5)
perl	Perl Programmable backend
shell	Shell (extern program) backend
sql	SQL Programmable backend
tcp	TCP Programmable backend

Esempio:

backend bdb

Ciò contrassegna l'inizio di nuova definizione backend BDB

3.4. Direttive generali del database

Le direttive contenute in questa sezione si applicano soltanto ai database in cui esse sono definite. Sono supportate da ogni tipo di database.

```
database <type>
```

Questa istruzione contrassegna l'inizio di nuova definizione di istanza di database. <type> dovrebbe essere uno dei tipi backend elencati al punto precedente.

Esempio:

database ldbm

Ciò contrassegna l'inizio di un detabase backend LDBM.

```
readonly { on | off }
```

Questa istruzione pone il database in modalità "read-only". Ogni tentativo di modificare il database restituirà l'errore "unwilling to perform".

Default:

readonly off

```
replica host=<hostname>[:<port>]
        [bindmethod={ simple | kerberos | sasl }]
        ["binddn=<DN>"]
        [mech=<mech>]
        [authcid=<identity>]
        [authzid=<identity>]
        [credentials=<password>]
        [srvtab=<filename>]
```

Questa direttiva specifica un luogo di replica per questo database. Il parametro uri= specifica uno schema, un host e facoltativamente una porta a cui l'istanza slave slapd può essere trovata. Un nome completo di dominio o un indirizzo IP può essere usato per l'<hostname>. Se <port> non è dato, viene usato il numero di porta standard di LDAP (389 o 636).

Il parametro bindnn= fornisce il DN con cui collegarsi allo slapd secondario per gli aggiornamenti. Dovrebbe essere un DN che ha accesso in lettura /scrittura al database dello slapd secondario, impostato tipicamente come un root dn nel file di configurazione dello slapd secondario. Deve anche accordarsi con la direttiva update dn nel file di configurazione dello slapd secondario. Generalmente questo DN *non deve* essere lo stesso del rootdn del master database. Poiché i DN contengono facilmente spazi, l'intera stringa "binddn=<DN>" deve essere racchiusa tra doppi apici.

L'istruzione bindmethod accetta come argometo i valori simple o Kerberos o sasl, a seconda che nella connessione allo slapd secondario si usi per l'autenticazione il metodo semplice basato su password o Kerberos o SASL.

L'autenticazione simple non dovrebbe essere usata a meno che non siano in funzione adeguate protezioni di segretezza e di integrità (per esempio TLS o IPSEC). L'autenticazione simple richiede la descrizione dettagliata di binddn e dei parametri delle credenziali.

L'autenticazione Kerberos è deprecata a favore dei meccanismi di autenticazione SASL, in particolare i meccanismi di GSSAPI e di KERBEROS_V4. L'autenticazione Kerberos richiede i parametri srvtab e binddn.

L'autenticazione SASL è generalmente raccomandata. L'autenticazione SASL richiede la specifica di un meccanismo usando il parametro saslmech. A seconda del meccanismo, un'identità e/o delle credenziali di autenticazione possono essere specificate usando authcid e le rispettive credenziali. Il parametro authzid può essere usato per specificare un'identità di autorizzazione.

Controllare questo URL per dettagli aggiuntivi: Replication with Slurpd (http://www.openldap.org/doc/admin/replication.html).

```
replogfile <filename>
```

Questa direttiva specifica il nome del file di log della replica in cui lo slapd registrerà i cambiamenti. Il log della replica è tipicamente scritto da slapd e letto da slurpd. Normalmente, questa direttiva è usata soltanto se slurpd è stato usato per replicare il database. Tuttavia, si può anche usare per generare un log di transazione, se lo slurpd non sta funzionando. In questo caso, dovrete troncare periodicamente il file, altrimenti crescerebbe indefinitamente.

Controllare questo URL per vedere dettagli addizionali: Replication with Slurpd (http://www.openldap.org/doc/admin/replication.html).

```
rootdn <dn>
```

Questa direttiva specifica il DN che non è soggetto a restrizioni di controllo di accesso o di limiti amministrativi per le operazioni su questo database. Il DN non ha bisogno riferirsi ad un oggetto nella directory. Il DN può riferirsi ad un'identità di SASL.

Entry-based Esempio:

rootdn "cn=Manager, dc=example, dc=com"

SASL-based Esempio:

rootdn "uid=root@EXAMPLE.COM"

```
rootpw <password>
```

Questa direttiva può essere usata per specificare una password per il rootdn (quando il rootdn è impostato a DN all'interno del database).

Esempio:

rootpw secret

È anche permesso fornire l'hash della password in forma RFC 2307. slappasswd può essere usato per generare l'hash della password.

Esempio:

rootpw {SSHA}ZKKuqbEKJfKSXhUbHG3fG8MDn9j1v4QN

L'hash è stato generato usando il comando slappasswd -s secret.

`suffix <dn suffix>`

Questa direttiva specifica il suffisso DN delle interrogazioni che saranno passate a questo database backend. Possono essere date più linee di suffisso e per ogni definizione di database è necessaria ce ne sia almeno una.

Esempio:

suffix "dc=example, dc=com"

Le interrogazioni con un DN che termina in "dc=example, dc=com" saranno passate a questo backend.

Nota: quando viene selezionato il backend a cui passare una interrogazione, lo slapd guarda alla/e linee di suffisso in ogni definizione del database nell'ordine che compaiono nel file. Quindi, se un suffisso di database è un prefisso di un altro, deve comparire dopo esso nel file di configurazione.

`syncrepl`

Questa direttiva può essere usata per tenere sincronizzato un database replicato con il database master, in modo che il contenuto del database replicato sia aggiornato con il contenuto del master.

Questo documento non copre in dettaglio questa direttiva, poiché si configura un server LDAP singolo. Per maggiori informazioni su questa direttiva, visitare: LDAP Sync Replication (http://www.openldap.org/doc/admin22/syncrepl.html).

`updatedn <dn>`

Questa direttiva è applicabile soltanto a uno slapd secondario. Essa specifica il DN a cui è permesso di apportare modifiche alla replica. Questo può essere il DN con cui si connette slurpd quando apporta modifiche alla replica o il DN associato con un'identità SASL.

Entry-based Esempio:

updatedn "cn=Update Daemon, dc=example, dc=com"

SASL-based Esempio:

updatedn "uid=slurpd@EXAMPLE.COM"

Controllare questo URL per maggiori dettagli: Replication with Slurpd (http://www.openldap.org/doc/admin/replication.html).

`updateref <URL>`

Questa istruzione è applicabile soltanto in uno slapd secondario. Essa Specifica l'URL da restituire ai client che inoltrano richieste di aggiornamento alla replica. Se specificato più volte, viene fornito ogni URL.

Esempio:

update ldap://master.example.net

3.5. Direttive del database BDB

Le direttive in questa categoria riguardano soltanto un database BDB. Cioè, esse devono seguire una linea del "database bdb" e venire prima di ogni successiva linea di "backend" o "database".

`directory <directory>`

Questa istruzione specifica la directory dove risiedono i file BDB contenenti il database e gli indici associati.

Default:

directory /usr/local/var/openldap-data

`sessionlog <sid> <limit>`

Questa direttiva specifica un archivio del log di sessione nel server del provider di replica syncrepl, che contiene informazioni sulle voci che sono state portate fuori dal contenuto di replica identificato da <sid>. La prima richiesta di ricerca syncrepl avente lo stesso valore <sid> nel cookie stabilisce l'archivio del log di sessione nel server del provider. Il numero di voci nell'archivio del log di sessione è limitato da <limit>. Le voci in eccesso sono rimosse dall'archivio in ordine FIFO. Sia <sid> che <limit> sono interi non negativi. <sid> ha non più di tre cifre decimali.

L'operazione di sincronizzazione del contenuto di LDAP che cade in una sessione preesistente può usare l'archivio del log di sessione allo scopo di ridurre l'ammontare di traffico di sincronizzazione. Se la replica non è troppo datata e può essere aggiornata dalle informazioni nell'archivio sessione, lo slapd provider invierà allo slapd consumer le identità delle voci eliminate assieme alle voci aggiunte o modificate nel contenuto di replicazione. Se lo stato della replica è troppo superato e oltre la copertura dell'archivio storico, allora lo slapd provider invierà le identità delle voci in ingresso non mutate assieme a quella delle voci cambiate. Lo slapd consumer quindi rimuoverà quelle voci nella replica che non sono identificate come presenti nel contenuto del provider.

Per maggiori informazioni su syncrepl, visitare: LDAP Sync Replication (http://www.openldap.org/doc/admin22/syncrepl.html).

3.6. Direttive del database LDBM

Le direttive contenute in questa categoria si applicano soltanto al database di backend LDBM. Cioè, esse devono seguire una linea "database ldbm" e venire prima di ogni altra linea di "database" o "backend".

```
cachesize <integer>
```

Questa direttiva specifica la quantità di oggetti per la memoria cache gestita dall'istanza di database backend LDBM.

Default:

cachesize 1000

```
dbcachesize <integer>
```

Questa istruzione specifica la dimensione in byte della memoria cache connessa associata ad ogni file indice aperto. Se non è supportata dal metodo del database sottostante, questa direttiva è ignorata senza commenti. Aumentando questo numero viene usata più memoria, ma questo può causare un notevole aumento di prestazioni, particolarmente durante modifiche o quando vengono costruiti gli indici.

Default:

dbcachesize 100000

```
dbnolocking
```

Questa opzione, se presente, disabilita il blocco del database. Abilitare questa opzione può migliorare le prestazioni a scapito della protezione dei dati.

```
dbnosync
```

Questa opzione fa si che i contenuti del database su disco non siano immediatamente sincronizzati con le continue modifiche in memoria. Abilitare questa opzione può migliorare le prestazioni a scapito della protezione dei dati.

```
directory <directory>
```

Questa istruzione specifica la directory dove risiedono i file LDBM contenenti il database e gli indici associati.

Default:

directory /usr/local/var/openldap-ldbm

```
index {<attrlist> | default} [pres,eq,approx,sub,none]
```

Questa direttiva specifica gli indici da mantenere per il dato attributo. Se viene fornito un solo <attrlist>, vengono mantenuti gli indici di default.

Esempio:

```
index default pres,eq
index uid
index cn,sn pres,eq,sub
index objectClass eq
```

La prima linea imposta l'insieme degli indici di default da mantenere per present ed equality. La seconda linea fa sì che l'insieme degli indici di default (pres, eq) sia mantenuto per il tipo di attributo uid. La terza linea fa sì che gli indici di present, equality e substring siano mantenuti per i tipi di attributo cn e sn. La quarta linea crea un indice equality per l'attributo di tipo objectClass.

Di default, non è mantenuto nessun indice. Generalmente si raccomanda come minimo di mantenere un indice equality sugli objectClass.

index objectClass eq

```
mode <integer>
```

Questa direttiva specifica i permessi di accesso che i file indice del database generato ex novo dovrebbero avere.

Default:

mode 0600

3.7. Esempi di Access Control

La funzione di access control fornita dalla direttiva *access* è abbastanza potente. Questa sezione mostra alcuni esempi di utilizzo. In primo luogo, alcuni esempi semplici:

```
access to * by * read
```

Questa direttiva di accesso garantisce accesso in lettura a tutti.

Il seguente esempio mostra l'uso di una espressione regolare per selezionare le voci per DN in due direttive di accesso dove l'ordinamento è importante.

```
access to dn=".*, o=U of M, c=US"
by * search
access to dn=".*, c=US"
by * read
```

L'accesso in lettura è assegnato alle voci sotto il sottoalbero c=US, tranne quelle voci sotto il sottoalbero ""o=U of M, c=US", a cui è garantito l'accesso in ricerca. Nessun accesso è assegnato ai c=US, perché nessuna direttiva di accesso corrisponde a questo DN. Se l'ordine di queste direttive di accesso fosse invertito, la direttiva specifica U-M non troverebbe mai corrispondenza, poiché tutti i campi U-m sono anche campi c=US.

Un altro modo di implementare lo stesso controllo di accesso è:

```
access to dn.children="dc=example,dc=com"
by * search
access to dn.children="dc=com"
by * read
```

L'accesso in lettura è consentito alle voci nel sottoalbero dc=com, tranne per quelle voci nel sottoalbero dc=example,dc=com, a cui è consentito accesso in ricerca. Non è consentito l'accesso a dc=com, poiché nessuna direttiva di accesso corrisponde a questo DN. Se l'ordine di queste direttive di accesso fosse invertito, la direttiva trascinata non verrebbe mai raggiunta, poiché tutte le voci sotto; dc=example,dc=com sono anche voci dc=com.

Nota: notare anche che se nessuna direttiva di accesso o nessuna clausola "by <who>" corrisponde, **l'accesso è negato**. Questo significa che ogni direttiva *access to* termina con una clausola implicita *by * none* , e ciascun elenco di accesso termina con una direttiva implicita *access to * by * none*.

L'esempio seguente mostra ancora l'importanza dell'ordinamento, sia delle direttive di accesso che delle clausole "by <who>". Inoltre mostra l'uso di un selettore di attributo per garantire l'accesso a un attributo specifico e vari selettori <who>.

```
access to dn.subtree="dc=example,dc=com" attr=homePhone
 by self write
 by dn.children=dc=example,dc=com" search
 by peername=IP:10\..+ read
 access to dn.subtree="dc=example,dc=com"
 by self write
 by dn.children="dc=example,dc=com" search
 by anonymous auth
```

Questo esempio riguarda le voci nel sottoalbero "dc=example,dc=com". A tutti gli attributi tranne homePhone, una voce può scrivere su se stessa, le voci nei campi example.com possono cercare da queste, nessun altro ha accesso (implicito by * none) tranne per autenticazione/autorizzazione (che è sempre fatta anonimamente). L'attributo homePhone è scrivibile dalla voce, ricercabile dai campi sotto example.com, leggibile dai client che si connettono dalla rete 10, e non è altrimenti leggibile (implicito by * none). Tutti gli altri accessi sono negati dall'implicito access to * by * none.

A volte è utile per consentire a un particolare DN di aggiungere o rimuovere se stesso da un attributo. Per esempio, se si volesse creare un gruppo di utenti e concedere loro di aggiungere e rimuovere soltanto il proprio DN dall'attributo membro, si potrebbe fare ciò tramite un'istruzione di accesso come questa:

```
access to attr=member,entry
        by dnattr=member selfwrite
```

Il selettore dnattr <who> comunica che l'accesso si applica agli oggetti elencati nell'attributo membro. Il selettore di accesso del selfwrite comunica che tali membri possono aggiungere o cancellare soltanto il loro proprio DN dall'attributo, e non altri valori. L'aggiunta del campo attributo è necessaria poiché l'accesso alla voce è necessario per accedere a qualunque attributo dell'oggetto.

C'è abbondanza di informazioni sui controllo di accesso sulla Guida dell'Amministratore OpenLDAP. Consultare: Access Control (http://www.openldap.org/doc/admin/slapdconfig.html#Access Control) per maggiori informazioni su questo argomento.

3.8. Esempio del file di configurazione

Quello che segue è un esempio di file di configurazione, suddiviso con testo esplicativo. Definisce due database per gestire le parti differenti dell'albero X.500; entrambe sono istanze di database BDB. I numeri della linea indicati sono forniti soltanto come riferimento e non sono inclusi nel file reale. In primo luogo, la sezione di configurazione globale:

```
1.    # example config file - global configuration section
2.    include /usr/local/etc/schema/core.schema
3.    referral ldap://root.openldap.org
4.    access to * by * read
```

La linea 1 è un commento. La linea 2 include un altro file di config il quale contiene le definizioni dello schema del nucleo. La direttiva di rinvio nella linea 3 significa che le domande non locali ad uno dei database definiti sotto si riferiranno al server LDAP che funziona sulla porta standard (389) all'host root.openldap.org.

La linea 4 è un controllo di accesso globale. Si applica a tutti i campi (dopo qualsiasi comando di accesso al database-specifico applicabile).

La sezione successiva del file di configurazione definisce un backend BDB che gestirà le domande per cose nella porzione dell'albero "dc=example,dc=com". Il database viene replicato su due slapd slave, una sui truelies, l'altra su judgmentday. Gli indici devono essere mantenuti per numerosi attributi e l'attributo userPassword deve essere protetto da accessi non autorizzati.

```
5.     # BDB definition for the example.com
6.     database bdb
7.     suffix "dc=example,dc=com"
8.     directory /usr/local/var/openldap-data
9.     rootdn "cn=Manager,dc=example,dc=com"
10.    rootpw secret
11.    # replication directives
12.    replogfile /usr/local/var/openldap/slapd.replog
13.    replica uri=ldap://slave1.example.com:389
14.            binddn="cn=Replicator,dc=example,dc=com"
15.            bindmethod=simple credentials=secret
16.    replica uri=ldaps://slave2.example.com:636
17.            binddn="cn=Replicator,dc=example,dc=com"
18.            bindmethod=simple credentials=secret
19.    # indexed attribute definitions
20.    index uid pres,eq
21.    index cn,sn,uid pres,eq,sub
22.    index objectClass eq
23.    # database access control definitions
24.    access to attr=userPassword
25.            by self write
26.            by anonymous auth
27.            by dn.base="cn=Admin,dc=example,dc=com" write
28.            by * none
29.    access to *
30.            by self write
31.            by dn.base="cn=Admin,dc=example,dc=com" write
32.            by * read
```

La linea 5 è un commento. L'inizio della definizione del database è contrassegnato dalla parola chiave del database alla linea 6. La linea 7 specifica il suffisso di DN per le domande da passare a questo database. La linea 8 specifica la directory in cui i file del database saranno presenti.

Le linee 9 e 10 identificano la voce "super user" del database e la password collegata. Questo campo non è soggetto a controllo di accesso o a limitazioni di scadenza o di formato. Ricordare di cifrare il rootpw usando slappasswd.

Esempio: rootpw{SSHA}Jq4xhhkGa7weT/0xKmaecT4HEXsdqiYA

Le linee da 11 a 18 sono per la replica. Vedere il link Replication (http://www.openldap.org/doc/admin/replication.html) per maggiori informazioni su queste direttive.

Le linee da 20 a 22 indicano gli indici da mantenere per i vari attributi.

Le linee da 24 a 32 specificano il controllo di accesso per i campi in questo database. Poiché questo è il primo database, i controlli si applicano anche ai campi non contenuti in alcun database (come la Root DSE). Per tutti i campi applicabili, l'attributo di userPassword è scrivibile dall'oggetto stesso e dall'oggetto "admin". Può essere usato per scopi di autenticazione/autorizzazione, ma non è altrimenti leggibile. Tutti gli altri attributi sono scrivibili dall'oggetto e dall'oggetto "admin" ma possono essere letti da tutti gli utenti (autenticati o no).

La sezione successiva dell'esempio del file di configurazione definisce un altro database BDB. Questo gestisce le domande che riguardano il sottoalbero dc=example,dc=net, ma è gestito dalla stessa entità del primo database. Si noti che senza la linea 39, l'accesso in lettura verrebbe concesso grazie alla regola globale di accesso contenuta nella linea 4.

```
33.    # BDB definition for example.net
34.    database bdb
35.    suffix "dc=example,dc=net"
36.    directory /usr/local/var/openldap-data-net
37.    rootdn "cn=Manager,dc=example,dc=com"
38.    index objectClass eq
39.    access to * by users read
```

Capitolo 4. Eseguire il server LDAP

Il demone *slapd* di LDAP è progettato per funzionare come server stand-alone. Questo permette al server di avere i vantaggi del caching, gestire i problemi di concorrenza con il database sottostante, e conservare risorse di sistema. Il funzionamento da inetd(8) non è previsto.

4.1. Opzioni della linea di comando

Slapd supporta un certo numero di opzioni della linea di comando, come dettagliato nella pagina di manuale. Questa sezione specifica alcune opzioni comunemente usate:

```
-f <filename>
```

Questa opzione specifica un file di configurazione alternativo per slapd. Il default è normalmente /usr/local/etc/openldap/slapd.conf.

```
-h <URLs>
```

Questa opzione specifica le configurazioni alternative dell' ascoltatore. Il default è ldap:/// che comporta LDAP sopra TCP su tutte le interfacce sulla porta LDAP di default 389. Si possono specificare le coppie host-porta o altri schemi di protocollo (quali ldaps:// o ldapi://). Per esempio, -h "ldaps:// ldap://127.0.0.1:667" genererà due ascoltatori: uno per LDAP sopra SSL su tutte le interfacce sulla porta 636 di default LDAP/SSL ed uno per LDAP sopra TCP sull'interfaccia del localhost (loopback) sulla porta 667. Gli host possono essere specificati usando il formato IPV4 punto-decimale usando i nomi di host. I valori delle porte devono essere numerici.

```
-n <service-name>
```

questa opzione specifica il nome del servizio usato per fare il log e per altri scopi. Il nome del servizio di default è slapd.

```
-l <syslog-local-user>
```

Questa opzione specifica l'utente locale per il servizio syslog(8). I valori possono essere LOCAL0, LOCAL1, LOCAL2... e LOCAL7. Il default è LOCAL4. Questa opzione potrebbe non essere supportata su tutti i sistemi. Vedere i la Sezione 6.5 per maggiori dettagli.

```
-u user -g group
```

Queste opzioni specificano rispettivamente l'utente e il gruppo necessari per fare funzionare lo slapd. L'utente può essere o un username o uid. Il gruppo può essere un nome gruppo o gid.

```
-r directory
```

Questa opzione specifica una directory run-time. Slapd eseguirà chroot(2) in questa directory dopo aver aperto gli ascoltatori, ma prima di leggere qualunque file di configurazione o di inizializzare qualunque backend.

```
-d <level> | ?
```

Questa opzione regola il livello di debug di slapd a <level>. Quando il livello è un carattere '?', i vari livelli di debug sono stampati e slapd esce, incurante di qualsiasi altra opzione data. I livelli correnti di debug sono:

Tabella 4-1. Livelli di debug

Livello	Descrizione
-1	abilita tutto il debug
0	nessun debug
1	traccia le chiamate alle funzioni
2	debug il packet handling
4	heavy trace debugging
8	gestione della connessione
16	stampa i pacchetti inviati e ricevuti
32	processamento del filtro di ricerca
64	processamento del file di configurazione
128	processamento lista di controllo accesso
256	stats log connections/operations/results
512	stats log entries sent
1024	stampa la comunicazione con i backend della shell
2048	print entry parsing debugging

Si possono permettere livelli multipli specificando l'opzione di debug ogni volta per ogni livello voluto. O, poiché i livelli di debug sono additivi, si possono fare i calcoli da sè. Quindi se si vogliono tracciare le chiamate alle funzioni e verificare che il file di configurazione venga processato si può impostare il livello come la somma di questi due livelli (in questo caso, - d 65). O si può lasciare che slapd faccia il calcolo, (per esempio - d 1 - d 64). Consultare <ldap.h>per maggiori dettagli.

Nota: lo slapd deve essere compilato con - DLDAP_DEBUG definito in modo che tutte le informazioni di debug oltre i due livelli di stat siano disponibili.

4.2. Avviare il server LDAP

in generale, slapd si avvia così:

```
/usr/local/etc/libexec/slapd [<option>]*
```

Dove/usr/local/etc/libexec è determinato per configurare e <option> è una delle opzioni descritte precedentemente (o in slapd(8)). A meno che si specifichi un livello di debug (incluso Livello 0), lo slapd automaticamente si biforcherà e si staccherà dal relativo terminale di controllo e girerà in background.

4.3. Terminare il server LDAP

Per terminare slapd in modo sicuro, si può usare questo comando:

```
kill -TERM `cat $(ETCDIR)/slapd.pid`
```

Terminare slapd con un metodo più drastico può causare la corruzione del database, perché potrebbe avere bisogno di svuotare i vari buffer prima di uscire. Si noti che lo slapd scrive il suo pid in un file denominato slapd.pid nella directory che si è configurato nel file di slapd.conf, per esempio:/usr/local/var/slapd.pid

Slapd inoltre scriverà i suoi argomenti in un file denominato slapd.args nella directory che si è configurato nel file slapd.conf, per esempio/usr/local/var/slapd.args

Capitolo 5. Creazione e manutenzione del database

Questa sezione spiega come generare un database slapd da zero. Ci sono due modi per creare un database. In primo luogo, si può creare il database on line usando LDAP . Con questo metodo si avvia semplicemente slapd e si aggiungono i campi usando il client LDAP di propria scelta. Questo metodo è valido per database relativamente piccoli (poche centinaia o migliaia di campi, secondo le proprie esigenze). Questo metodo funziona per i tipi di database che supportano gli aggiornamenti.

Il secondo metodo di creazione del database è farlo off-line usando programmi di utilità speciali forniti da slapd. Questo metodo è il migliore se si devono creare molte migliaia di campi, che richiederebbe un tempo inaccettabilmente lungo usando il metodo LDAP, o se si desidera assicurare l'inaccessibilità del database in fase di creazione. Si noti che non tutti i tipi di database supportano queste utilitità.

5.1. Creazione di un database on line

ll pacchetto di programmi OpenLDAP, è munito di una utilità denominata ldapadd, usata per aggiungere i campi durante il funzionamento del server LDAP. Se si sceglie di creare il Database on line, si può utilizzare lo strumento ldapadd per aggiungere i campi (si possono anche usare altri client forniti fuori dal pacchetto OpenLDAP per aggiungere i campi, come il Browser Ldap (http://www.iit.edu/~gawojar/ldap/)). Dopo l'aggiunta dei primi campi, si può ancora usare ldapadd per aggiungerne ancora. Bisogna essere sicuri di aver messo le seguenti opzioni di configurazione sul proprio file sladp.conf prima di far partire slapd:

```
suffix <dn>
```

Come descritto nella la Sezione 3.4, questa opzione comunica quali campi devono essere tenuti da questo database. Bisogna impostare questo al DN della radice del sotto-albero che si sta provando a generare. Per esempio:

```
suffix "o=TUDelft, c=NL"
```

Si dovrebbe essere sicuri di specificare una directory in cui i file di indice verranno creati:

```
directory /usr/local/tudelft
```

Bisogna generare questa directory con i permessi appropriati, così che lo slapd possa scrivere in essa.

Bisogna configurare slapd in modo da poterlo connettere come utente della directory con permesso di aggiungere oggetti. Si può configurare la directory per supportare uno speciale super-utente o utente root, proprio per questo fine. Ciò è fatto con le seguenti due opzioni nella definizione del database:

```
rootdn <dn>
rootpw <passwd>   /* Remember to use a SHA password here !!! */
```

Queste opzioni specificano un DN e una password che possono essere usati per autenticarsi come campo "superutente" nel database (cioè, l'oggetto che può fare tutto). Il DN e la password specificati qui funzioneranno sempre, senza verificare se l'oggetto chiamato realmente esiste o abbia la password data. Questo risolve il problema dell'uovo e della gallina , cioè quello di autenticare ed aggiungere gli oggetti prima che tutti gli oggetti tuttavia esistano.

Slapd capisce nativamente se si usa una password cifrata SHA-1 sulla direttiva rootpw. Io uso una classe Java che genera le password SHA-1, ma è possibile usare il comando *slappasswd* per generare le password:

```
slappasswd -h {SHA}

rootpw    "{SHA}5en6G6MezRroT3XKqkdPOmY/BfQ="
```

Per esempio:

```
        rootdn "cn=Manager,dc=example,dc=com"
        rootpw "{SHA}5en6G6MezRroT3XKqkdPOmY/BfQ="
```

L'output di default per slappasswd è di generare password Secure Hash {SSHA}, in questo caso non è necessario passare il parametro -h, solo chiamare direttamente slappasswd.

Se si sta usando SASL come meccanismo di autenticazione verso LDAP, la linea del rootpw può essere scartata. Dare uno sguardo sulla la Sezione 3.4 e sulla la Sezione 6.2 per maggiori dettagli.

Per concludere, dovreste assicurarvi che la definizione di database contenga le definizioni di indice desiderate:

```
index {<attrlist> | default} [pres,eq,sub,none]
```

Per esempio, per indicizzare gli attributi cn, sn, uid e objectclass, possono essere usate le seguenti linee di configurazione dell'indice.

```
index cn,sn,uid pres,eq,sub
index objectClass pres,eq
```

Nota: Si noti che non tutti i tipi di indice sono disponibili con tutti i tipi di attributo. Si dia uno sguardo su la Sezione 3.6 per gli esempi.

Una volta che si sono configurate le cose a proprio gradimento, avviare slapd, collegarsi con il proprio client LDAP, e iniziate ad aggiungere gli oggetti. Per esempio, per aggiungere il campo TUDelft seguito da un campo postmaster per mezzo dello strumento ldapadd, si può creare un file denominato /tmp/newentry contenente:

```
o=TUDelft, c=NL
objectClass=organization
description=Technical University of Delft Netherlands

cn=Postmaster, o=TUDelft, c=NL
objectClass=organizationalRole
cn=Postmaster
description= TUDelft postmaster - postmaster@tudelft.nl
```

e poi usare un comando come questo per creare realmente l'oggetto:

```
ldapadd -f /tmp/newentry -x -D "cn=Manager, o=TUDelft, c=NL" -w secret
```

Il precedente comando suppone che si sia impostato il rootdn a "cn=Manager, o=TUDelft, c=NL" e rootpw a "segreto" (forse SHA-1 cifrato su slapd.conf). Se non si vuole scrivere la password sulla linea di comando, usare l'opzione -W per il comando ldapadd invece di -w "password". Comparirà la richiesta di digitare la password:

```
ldapadd -f /tmp/newentry -x -D "cn=Manager, o=TUDelft, c=NL" -W
Enter LDAP Password:
```

5.2. Creare un database off line

Il secondo metodo di creazione del database è farlo off-line, usando gli strumenti del database di slapd descritti qui di seguito. Questo metodo è il migliore se avete molte migliaia di oggetti da produrre, in quanto con il primo metodo occorrerebbe una quantità di tempo inaccettabile cioè usando il metodo LDAP descritto precedentemente. Questi strumenti leggono il file di configurazione dello slapd e il file dell'input LDIF che contengono una rappresentazione del testo degli oggetti da aggiungere. Producono direttamente i file index del database. Ci sono parecchie opzioni di configurazione importanti delle quali vorreste essere sicuri di impostare nei config file del database in primo luogo:

```
suffix <dn>
```

Come descritto nella sezione precedente, questa opzione comunica quali oggetti devono essere tenuti da questo database. Si dovrebbe impostare questo al DN della radice del sotto-albero che si sta provando a creare. Per esempio:

```
suffix "o=TUDelft, c=NL"
```

Si dovrebbe essere sicuri di specificare una directory in cui i file index dovrebbero essere creati:

```
directory /usr/local/tudelft
```

Per concludere, si dovrebbe specificare quali indici si desidera sviluppare. Ciò è fatto da una o più opzioni di indice.

```
index {<attrlist> | default } [pres,eq,approx,sub,none]
```

Per esempio:

```
index cn,sn,uid pres,eq,sub
index objectClass eq
```

Ciò genererebbe gli indici di presenza, di uguaglianza e di sottostringa per il cn, Sn ed gli attributi del uid e un indice di uguaglianza per l'attributo objectClass. Vedere la sezione file di configurazione per maggiori informazioni su questa opzione.

Una volta che si sono configurate le cose a proprio gradimento, creare il database primario e gli indici collegati facendo funzionare il programma slapadd(8):

```
slapadd -l <inputfile> -f <slapdconfigfile> [-d <debuglevel>]
 [-n <integer>|-b <suffix>]
```

Gli argomenti hanno i seguenti significati:

```
-l <inputfile>
```

Specifica il file input di LDIF contenente gli oggetti da aggiungere nella forma del testo (dare uno sguardo alla prossima sezione).

```
-f <slapdconfigfile>
```

Specifica il file di configurazione dello slapd che comunica dove creare gli indici, che indici creare, ecc.

```
-d <debuglevel>
```

Attiva il debbuging, come specificato da <debuglevel>. I livelli di debug sono gli stessi come per slapd. Vedere la la Sezione 4.1 per maggiori dettagli.

```
-n <databasenumber>
```

Un argomento opzionale che specifica quale database modificare. Il primo database elencato nel file di configurazione è 1, i secondi 2, ecc. Di default è usato il primo database nel file di configurazione. Non dovrebbe essere usato insieme a - b.

```
-b <suffix>
```

Un argomento opzionale che specifica quale database modificare. Il suffisso fornito è abbinato a un suffisso di una direttiva del database per determinare il numero del database. Non dovrebbe essere insieme ad - n.

A volte può essere necessario rigenerare gli indici (come dopo aver modificato slapd.conf(5)). Ciò è possibile usando il programma slapindex(8). Il programma slapindex è invocato in questo modo:

```
slapindex -f <slapdconfigfile> [-d <debuglevel>] [-n <databasenumber>|-b <suffix>]
```

Dove le opzioni - f, - d, - n e -b sono le stesse che per il programma slapadd(1). slapindex ricostruisce tutti gli indici basati sui contenuti correnti del database.

Il programma slapcat è usato per fare il dump del database in un file LDIF. Ciò può essere utile quando si desidera fare un backup leggibile del proprio database o quando si desidera editare il proprio database off-line. Il programma è invocato in questo modo:

```
slapcat -l <filename> -f <slapdconfigfile> [-d <debuglevel>] [-n <databasenumber>|-b <su
```

dove -n o -b si usano per selezionare il database nello slapd.conf(5) specificato usando -f. L'uscita corrispondente di LDIF è scritta sullo standard output o in un determinato file usando l'opzione -l.

5.3. Maggiori informazioni sul formato LDIF

L'LDAP Data Interchange Format (LDIF) è usato per rappresentare i campi di LDAP in un formato testo semplice. La forma di base di un campo è:

```
#comment
dn: <distinguished name>
<attrdesc>: <attrvalue>
<attrdesc>: <attrvalue>
...
```

Le linee che cominciano con un '#' sono commenti. Un attributo di descrizione (attrdesc) può essere un tipo semplice di attributo come cn o objectClass o 1, 2, 3 (un OID associato con un tipo di attributo) o può includere opzioni quali i cn; lang_en_US o userCertificate; binary.

Una linea può essere continuata iniziando la linea seguente con un singolo spazio o un carattere tab. Per esempio:

```
dn: cn=Barbara J Jensen, dc=example, dc=
 com
cn: Barbara J
    Jensen
```

è equivalente a:

```
dn: cn=Barbara J Jensen, dc=example, dc=com
cn: Barbara J Jensen
```

I valori multipli di attributo sono specificati su linee separate. Per esempio:

```
cn: Barbara J Jensen
cn: Babs Jensen
```

Se un <attrvalue> contiene i caratteri non stampabili o comincia con uno spazio, un carattere due punti (':'), o un segno di minore,('<'), il <attrdesc> è seguito da due punti e dalla codifica base64 del valore. Per esempio, il valore "begins with a space" sarebbe codificato in questo modo:

```
cn:: IGJlZ2lucyB3aXRoIGEgc3BhY2U=
```

Si può anche specificare un URL che contiene il valore di attributo. Per esempio, ciò che segue specifica il valore jpegPhoto che si dovrebbe essere ottenere dal file /path/to/file.jpeg.

```
cn:< file://path/to/file.jpeg
```

Gli oggetti multipli all'interno dello stesso file LDIF sono separati da linee vuote. Qui c'è un esempio di un file LDIF che contiene tre campi.

```
# Barbara's Entry
dn: cn=Barbara J Jensen, dc=example, dc=com
cn: Barbara J Jensen
cn: Babs Jensen
objectClass: person
sn: Jensen

# Bjorn's Entry
dn: cn=Bjorn J Jensen, dc=example, dc=com
cn: Bjorn J Jensen
cn: Bjorn Jensen
objectClass: person
sn: Jensen
# Base64 encoded JPEG photo
jpegPhoto:: /9j/4AAQSkZJRgABAAAAAQABAAD/2wBDABALD
A4MChAODQ4SERATGCgaGBYWGDEjJR0oOjM9PDkzODdASFxOQ
ERXRTc4UG1RV19iZ2hnPk1xeXBkeFxlZ2P/2wBDARESEhgVG

# Jennifer's Entry
dn: cn=Jennifer J Jensen, dc=example, dc=com
cn: Jennifer J Jensen
cn: Jennifer Jensen
objectClass: person
sn: Jensen
# JPEG photo from file
jpegPhoto:< file://path/to/file.jpeg
```

Notare che il jpegPhoto nel campo Bjorn è codificato base 64 ed il jpegPhoto nell'oggetto Jennifer è ottenuto dalla posizione indicata dal URL.

Gli spazi posteriori non sono ripuliti dai valori in un file LDIF. Nè gli spazi interni multipli sono compressi. Se non sono desiderati nei propri dati, non li si metta.

5.4. Le utilità ldapsearch, ldapdelete e ldapmodify

ldapsearch - ldapsearch è un'interfaccia di shell che permette di accedere alla chiamata di libreria ldap_search(3). Usare questo programma di utilità per cercare i campi nel proprio database backend LDAP.

La sintassi per richiamare ldapsearch è la seguente (guardare alla man page del ldapsearch per vedere che cosa significa ogni opzione):

```
ldapsearch  [-n]  [-u]  [-v]  [-k]
[-K]  [-t]  [-A] [-B] [-L]
[-R] [-d debuglevel] [-F sep] [-f file]
[-x] [-D binddn]  [-W]  [-w bindpasswd]
[-h ldaphost]  [-p ldapport]   [-b searchbase]
[-s base|one|sub]
[-a never|always|search|find] [-l timelimit]
[-z sizelimit] filter [attrs...]
```

ldapsearch apre un collegamento con un server LDAP, bind, ed effettua una ricerca usando il filtro *filter*. Il filtro dovrebbe essere conforme alla rappresentazione della stringa per i filtri LDAP come definiti in RFC 1558. Se ldapsearch trova uno o più campi, gli attributi specificati da *attrs* sono richiamati e i campi ed i valori sono stampati sullo standard output. Se nessun attrs è elencato, tutti gli attributi sono restituiti.

```
ldapsearch -x -b 'o=TUDelft,c=NL' 'objectclass=*'

ldapsearch -b 'o=TUDelft,c=NL' 'cn=Rene van Leuken'

ldasearch -u -b 'o=TUDelft,c=NL' 'cn=Luiz Malere' sn mail
```

L'opzione -b sta per searchbase (punto iniziale di ricerca), l'opzione -u corrisponde alle informazioni di output userfriendly e l'opzione -x è usata per specificare la semplice autenticazione.

ldapdelete - ldapdelete è un'interfaccia accessibile dalla shell alla chiamata di libreria ldap_delete(3). Usare questo programma di utilità per eliminare gli oggetti sul proprio database backend Ldap.

La sintassi per richiamare ldapdelete è la seguente (dare un'occhiata alla man page del ldapdelete per vedere che cosa significa ogni opzione):

```
ldapdelete   [-n]   [-v]  [-k]  [-K]
[-c]  [-d debuglevel]  [-f file]  [-D binddn]
[-W]  [-w passwd] [-h ldaphost] [-p ldapport]
[dn]...
```

ldapdelete apre un collegamento con un server LDAP, bind, e cancella una o più campi. Se uno o più argomenti dn sono forniti, gli oggetti con quei Distinguished Names vengono eliminati. Ciascun DN dovrebbe essere rappresentato da una stringa DN come definito in RFC 1779. Se non sono forniti argomanti sul DN, una lista di DN viene letta dallo standard input (o dal file se è usato -f flag).

Qui ci sono alcuni esempi di uso di ldapdelete:

```
ldapdelete 'cn=Luiz Malere,o=TUDelft,c=NL'

ldapdelete -v 'cn=Rene van Leuken,o=TUDelft,c=NL' -D 'cn=Luiz Malere,o=TUDelft,c=NL' -W
```

L'opzione -v corrisponde al modo prolisso, l'opzione -D corrisponde a Binddn (il DN su cui fare l'autenticazione) e l'opzione -W corrisponde alla prompt della password.

ldapmodify - ldapmodify è un'interfaccia accessibile dalla shell che permette l'accesso alle chiamate di libreria ldap_modify(3) e ldap_add(3). Usare questo programma di utilità per modificare i campi sul proprio database backend LDAP.

La sintassi per richiamare ldapmodify è la seguente (dare un'occhiata alla man page di ldapmodify per vedere il significato di ogni opzione):

```
ldapmodify   [-a]  [-b]  [-c]  [-r]
[-n]  [-v]  [-k]  [-d debuglevel]
[-D binddn]  [-W]  [-w passwd]
[-h ldaphost] [-p ldapport] [-f file]

ldapadd [-b] [-c] [-r] [-n]
[-v]  [-k]  [-K]  [-d debuglevel]
[-D binddn]  [-w passwd]  [-h ldaphost]
[-p ldapport] [-f file]
```

ldapadd è implementato come un link fisico allo strumento ldapmodify. Quando invocato come ldapadd, il flag -a (aggiunge un nuovo campo) di ldapmodify è attivato automaticamente. ldapmodify apre un collegamento con un server LDAP, bind, e modifica o aggiunge voci. Le informazioni sugli oggetti sono lette dallo staqndard input o dal file attraverso l'uso dell'opzione -f.

Qui ci sono alcuni esempi sull'uso di ldapmodify:

Supponendo che il file /tmp/entrymods esista ed abbia i contenuti:

```
dn: cn=Modify Me, o=University of Michigan, c=US
changetype: modify
replace: mail
mail: modme@terminator.rs.itd.umich.edu
-
add: title
title: Grand Poobah
-
add: jpegPhoto
jpegPhoto: /tmp/modme.jpeg
-
delete: description
-
```

Il comando:

```
ldapmodify -b -r -f /tmp/entrymods
```

sostituirà i contenuti dell'attributo della voce dell'email "Modify Me" con il valore "modme@terminator.rs.itd.umich.edu", aggiungerà un titolo "Grand Poobah", e i contenuti del file /tmp/modme.jpeg come un jpegPhoto e rimuoverà completamente l'attributo descrizione.

Le stesse modifiche come sopra possono essere fatte usando il più vecchio formato di input ldapmodify:

```
cn=Modify Me, o=University of Michigan, c=US
mail=modme@terminator.rs.itd.umich.edu
+title=Grand Poobah
+jpegPhoto=/tmp/modme.jpeg
-description
```

E più il seguente comando:

```
ldapmodify -b -r -f /tmp/entrymods
```

Supponendo che il file /tmp/newentry esista ed abbia i contenuti:

```
dn: cn=Barbara Jensen, o=University of Michigan, c=US
objectClass: person
cn: Barbara Jensen
cn: Babs Jensen
sn: Jensen
title: the world's most famous manager
mail: bjensen@terminator.rs.itd.umich.edu
uid: bjensen
```

Il comando:

```
ldapadd -f /tmp/entrymods
```

aggiungerà il campo con dn: cn=Barbara Jensen, o=University del Michigan, c=US se non è già presente. Se un oggetto con questo dn esiste già, il comando indicherà l'errore e non sovrascriverà l'oggetto.

Supponendo che il file /tmp/newentry esista e abbia i contenuti:

```
dn: cn=Barbara Jensen, o=University of Michigan, c=US
changetype: delete
```

Il comando:

```
ldapmodify -f /tmp/entrymods
```

rimuoverà l'oggetto Babs Jensen.

L'opzione -f corrisponde al file (legge le informazioni di modifica da un file invece che dallo standard input), l'opzione -b corrisponde al binario (tutti i valori che cominciano con un '/' nel file di input sono interpretati come binari), -r corrisponde a 'replace' (sostituite i valori esistenti con il default).

Capitolo 6. Caratteristiche e informazioni aggiuntive

In questa sezione si troveranno informazioni addizionali e riferimenti utili per argomenti come Autenticazione, Log e client Ldap. Alla fine della sezione ci sono anche delle URL generiche molto buone e qualche libro riguardo l'argomento LDAP.

6.1. LDAP Migration Tools

Gli LDAP Migration Tools sono una collezione di Script Perl forniti da PADL software Ltd. Sono usati per convertire i file di configurazione nel formato LDIF. Si raccomanda di leggere i termini di licenza prima di usarli, anche se sono liberi. Se si progetta di usare il proprio server LDAP per autenticare gli utenti, questi strumenti potranno essere molto utili. Usare i Migration Tools per convertire il proprio NIS o i propri archivi di password nel formato LDIF, rendendo questi file compatibili con il proprio Server LDAP. Applicare anche questi script Perl per la migrazione di utenti, gruppi, alias, host, netgroup, reti, protocolli, RPC e servizi dai nameservice esistenti (NIS, file flat e NETinfo) al formato LDIF.

Per scaricare gli LDAP Migration Tools e avere maggiori informazioni, andare al seguente indirizzo: http://www.padl.com/tools.html.

Il pacchetto si presenta con un file README e il nome dei file di script sono intuitivi. Dare uno sguardo al file README e poi avviare applicando gli script.

Un altro URL raccomandato per i Migration Tool è:

http://dataconv.org/apps_ldap.html

6.2. Autenticazione usando LDAP

Per accedere al servizio LDAP, il primo client LDAP deve autenticare se stesso al servizio. Cioè, deve comunicare al server LDAP chi sta cercando di accedere ai dati così che il server puossa decidere cosa il client può vedere e fare. Se il client si autentica con successo al server LDAP, allora quando il server riceve successivamente una richiesta dal client, esso controllerà se al client è permesso fare (operare) la richiesta (domanda). Questo processo è chiamato controllo di accesso.

In LDAP, l'autenticazione è fornita nell'operazione "bind". Ldapv3 supporta tre tipi di autenticazioni: autenticazione anonima, semplice e SASL. Un client che trasmette una richiesta a LDAP senza fare un "bind" viene trattato come un client anonimo. L'autenticazione semplice consiste nel trasmettere al server LDAP il fully qualified DN del client (utente) e la password del client in chiaro. Questo meccanismo presenta problemi di sicurezza, perché la password può essere letta dalla rete. Per evitare di esporre la password in questo modo si può usare il meccanismo di autenticazione semplice all'interno di un canale cifrato (quale SSL), a condizione che questo sia fornito dal server LDAP.

Per concludere, SASL è il Simple Authentication and Security Layer (RFC 2222). Esso specifica un protocollo di domanda-risposta in cui i dati sono scambiati fra il client e il server per gli scopi dell'autenticazione e di instaurazione di uno strato di sicurezza su cui effettuare comunicazione successiva. Usando SASL, LDAP può supportare qualunque tipo di autenticazione gradito al client e al server LDAP. Il pacchetto di Cyrus-SASL è disponibile al seguente URL: http://asg.web.cmu.edu/sasl/sasl-library.html.

Oltre ad autenticare gli utenti per accedere alle informazioni dal proprio albero di directory, il proprio server LDAP può autenticare gli utenti da altri servizi (Sendmail, login, ftp, ecc.). Questo è ottenuto migrando informazioni specifiche dell'utente sul proprio server LDAP e usando un meccanismo denominato PAM (Pluggable Authentication Module). Il modulo di autenticazione per LDAP è disponibile come file di archivio in formato tar al seguente indirizzo: http://www.padl.com/OSS/pam_ldap.html

6.3. Configurazione di SASL : Digest - MD5

Sono riuscito a far funzionare l'autenticazione LDAP-SASL usando il meccanismo DIGEST-MD5. Per ottenere questo, ho seguito rigorosamente i passi elencati sotto:

- Scaricato SleepyCat 4.1.25, compilato e costruito manualmente. Dopo averlo scaricato, ho seguito le istruzioni elencate sul file doc/install.html sotto la directory in cui ho disimballato il pacchetto del tar.gz.

  Dopo il disimballaggio potete sperimentare il consiglio dato:

  ```
  root@rdnt03:/usr/local/db-4.1.25/build_unix#../dist/configure
  root@rdnt03:/usr/local/db-4.1.25/build_unix#make
  root@rdnt03:/usr/local/db-4.1.25/build_unix#make install
  ```

- Scaricato Cyrus SASL 2.1.12, disimballando e seguendo le istruzioni elencate sul documento doc/install.html, sotto la directoryt in cui ho disimballato il file del tar.gz. a questo punto è necessaria un po' di attenzione, dovete far funzionare lo script di configurazione usando alcuni parametri del env:

  ```
  root@rdnt03:/usr/local/cyrus-sasl-2.1.12#env CPPFLAGS="-I/usr/local/BerkeleyDB.4.1/incl
  LDFLAGS="-L/usr/local/BerkeleyDB.4.1/lib" ./configure
  ```

  I parametri ambientali CPPFLAGS e LDFLAGS mirerebbero rispettivamente alle directori include e lib dove è stato installato berkeley BDB.

  A questo punto potete far funzionare ciò che vi è stato suggerito:

```
root@rdnt03:/usr/local/cyrus-sasl-2.1.12#make
root@rdnt03:/usr/local/cyrus-sasl-2.1.12#make install
root@rdnt03:/usr/local/cyrus-sasl-2.1.12#ln -s /usr/local/lib/sasl2 /usr/lib/sasl2
```

- Per concludere, ho installato OpenLDAP 2.1.16 usandole stesse procedure elencate su questo documento, e facendo funzionare lo script di configurazione nello stesso modo della SASL configuration:

  ```
  root@rdnt03:/usr/local/openldap-2.1.16#env CPPFLAGS="-I/usr/local/BerkeleyDB.4.1/includ
  LDFLAGS="-L/usr/local/BerkeleyDB.4.1/lib" ./configure
  ```

  Dopo questo, ho attuato il suggerimento:

  ```
  root@rdnt03:/usr/local/openldap-2.1.16#make depend
  root@rdnt03:/usr/local/openldap-2.1.16#make
  root@rdnt03:/usr/local/openldap-2.1.16#make install
  ```

- Successivamente, ho creato l'user database di SASL:

  ```
  root@rdnt03:~# saslpasswd2 -c admin
  ```

  Sarete richiamati per inserire una password. Ricordarsi che l'username non dovrebbe essere un DN (nome distinto). Inoltre ricordarsi di usare la stessa password di administrator sull'albero della directory.

- Ora, dovreste regolare la direttiva sasl-regexp nel file *slapd.conf* prima di far partire il daemon slapd e di verificare l'autenticazione. Il mio file *slapd.conf* risiede in /usr/local/etc/openldap:

  ```
  sasl-regexp uid=(.*),cn=rdnt03,cn=DIGEST-MD5,cn=auth uid=$1,ou=People,o=Ever
  ```

  Questo parametro è nel formato di:

  uid=<username>,cn=<realm>,cn=<mech>,cn=auth

  Lo username è preso da sasl ed inserito nella stringa di ricerca di ldap nella posizione $1. Il proprio regno si presuppone essere il proprio FQDN (Fully Qualified Domain Name), ma in alcuni casi non è così, come nel mio. Si scopra che cosa è il proprio regno:

  ```
  root@rdnt03:~# sasldblistusers2
  admin@rdnt03: userPassword
  admin@rdnt03: cmusaslsecretOTP
  ```

  Nel mio caso, *rdnt03* è indicato come il regno. Se esso è il proprio FQDN non bisognerebbe avere problemi. Io uso il seguente file LDIF:

  ```
  dn: o=Ever
  o: Ever
  description: Organization Root
  objectClass: top
  objectClass: organization
  ```

```
dn: ou=Staff, o=Ever
ou: Staff
description: These are privileged users that can interact with Organization products
objectClass: top
objectClass: organizationalUnit

dn: ou=People, o=Ever
ou: People
objectClass: top
objectClass: organizationalUnit

dn: uid=admin, ou=Staff, o=Ever
uid: admin
cn: LDAP Adminstrator
sn: admin
userPassword: {SHA}5en6G6MezRroT3XKqkdPOmY/BfQ=
objectClass: Top
objectClass: Person
objectClass: Organizationalperson
objectClass: Inetorgperson

dn: uid=admin,ou=People,o=Ever
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
userPassword: {SHA}5en6G6MezRroT3XKqkdPOmY/BfQ=
displayName: admin
mail: admin@eversystems.com.br
uid: admin
cn: Administrator
sn: admin
```

Inserire i campi nella propria directory LDAP usando il seguente comando:

```
slapadd -c -l Ever.ldif -f slapd.conf -v -d 256
```

- Ora, avviare il demone *slapd* e lanciare una query usando il comando *ldapsearch*:

```
root@rdnt03:~# ldapsearch -U admin@rdnt03 -b 'o=Ever' '(objectclass=*)'
SASL/DIGEST-MD5 authentication started
Please enter your password:
SASL username: admin@rdnt03
SASL SSF: 128
SASL installing layers
...
Entries
...
```

Questo è tutto! Se si preferisce usare SASL con kerberos V o GSSAPI, c'è un link utile su http://www.openldap.org/doc/admin22/sasl.html. Questo link presuppone che si sia riusciti a installare e configurare la libreria di SASL. Le mailing list aiuteranno ad ottenere suggerimenti su questo problema: http://asg.web.cmu.edu/sasl/index.html#mailinglists

6.4. Strumenti grafici di LDAP

Kldap è un client grafico di LDAP scritto per KDE. Kldap ha un'interfaccia piacevole e può mostrare tutto l'albero delle informazioni immagazzinato sulla propria directory. Si possono vedere alcuni screenshot dell'applicazione e scaricarli presso: http://www.mountpoint.ch/oliver/kldap/

KDirAdm è uno strumento di amministrazione delle directory LDAP scritto per l'ambiente desktop KDE versione 2 o successiva. Mira a fornire tutte la funzionalità della maggior parte degli strumenti commerciali di amministrazione della directory: http://www.carillonis.com/kdiradm/

Directory Administrator è l'applicazione GNOME più utilizzata per la gestione di utenti e gruppi UNIX sui server directory LDAP. Gli amministratori di directory permettono di creare ed eliminare utenti e gruppi e di gestire le informazioni della rubrica degli indirizzi associata agli utenti, gestire il controllo di accesso e gli instradamenti di posta di Sendmail: http://diradmin.open-it.org/index.php

GQ è un altro client grafico di LDAP con un'interfaccia più semplice. È stato scritto per GNOME. Inoltre funziona sotto KDE, allo stesso modo come Kldap funziona sotto GNOME. L'indirizzo per scaricarlo e ottenere maggiori informazioni è: http://biot.com/gq/

LDAP Browser/Editor: Questo strumento è fantastico, ha funzionalità complete di navigazione e amministrative. Verificare: Ldap Browser (http://www.iit.edu/~gawojar/ldap/).

6.5. Log

Slapd usa il servizio syslog(8) per generare i log. L'utente predefinito del servizio syslog(8) è LOCAL4, ma i valori da LOCAL0, LOCAL1, fino a LOCAL7 sono permessi.

Per permettere la generazione dei logs bisogna editare il proprio file syslog.conf, situato solitamente nella directory /etc.

Create una linea come questa:

```
local4.*     /usr/adm/ldaplog
```

Questo userà di default l'utente LOCAL4 per il servizio syslog. Se non si ha familiarità con la sintassi di questa linea, dare un'occhiata alle man pages di syslog, syslog.conf e syslogd. Se si vuole specificare il livello dei log generati o cambiare l'utente predefinito, si hanno le seguenti opzioni durante l'avvio di slapd:

```
-s syslog-level
```

Questa opzione comunica allo slapd a quale livello di debugging dovrebbe collegarsi il servizio syslog(8). Il livello descrive la severità del messaggio ed è una parola chiave dalla seguente lista ordinata (dal maggiore al minore): emerg, alert, crit, err, warning, notice, info e debug. Es: slapd -f myslapd.conf -s debug

`-l syslog-local-user`

Selezionare l'utente locale per il servizio syslog(8). I valori possono essere LOCAL0, LOCAL1 e così via, fino a LOCAL7. Il default è LOCAL4. Tuttavia, questa opzione è consentita soltanto su sistemi che supportano gli utenti locali con il servizio syslog(8).

Ora dare un'occhiata ai log generati (/usr/adm/ldaplog nell'esempio). Possono aiutare immensamente nel risolvere problemi con domande, aggiornamenti, binding, ecc.

Capitolo 7. Riferimenti

In questa sezione si trova la documentazione supplementare su LDAP: URL utili, libri tosti e definizione di RFC.

7.1. URL

Qui ci sono URL contenenti informazioni molto utili su LDAP. Questo HOWTO è stato scritto utilizzando questi URL, quindi se dopo aver letto questo documento si ha bisogno di informazioni più specifiche, probabilmente saranno qui:

- University of Michigan LDAP Page: http://www.umich.edu/~dirsvcs/ldap/
- University of Michigan LDAP Documentation Page: http://www.umich.edu/~dirsvcs/ldap/doc/
- OpenLDAP Administrator's Guide (brother document): http://www.openldap.org/doc/admin
- Linux Directory Service: http://www.rage.net/ldap/
- Red Hat and LDAP: http://www.redhat.com/docs/manuals/linux/RHL-9-Manual/ref-guide/ch-ldap.html
- Mandrake Linux - Using OpenLDAP for Authentication: http://www.mandrakesecure.net/en/docs/ldap-auth.php
- Integrating OpenLDAP with other Open Source projects: ftp://kalamazoolinux.org/pub/pdf/ldapv3.pdf

7.2. Libri

Questi sono i libri più popolari e più utili su LDAP:

- Implementing LDAP di Mark Wilcox
- LDAP: Programming Directory-Enabled Applications with Lightweight Directory Access Protocol di Howes e Smith
- Understanding and Deploying LDAP Directory Servers di Howes, Smith, e Good

7.3. RFC

Gli RFC (http://www.rfc-editor.org/rfc/) che suportano gli sforzi di sviluppo di LDAP:

- RFC 1558: Una rappresentazione della stringa dei filtri di ricerca di LDAP

- RFC 1777: Directory Access Protocol Leggero
- RFC 1778: La rappresentazione della stringa delle sintassi standard di attributo
- RFC 1779: Una rappresentazione della stringa dei D N
- RFC 1781: Uso della directory OSI per realizzare chiamate user friendly
- RFC 1798: LDAP Connectionless
- RFC 1823: L'Interfaccia Di Programmazione e Di Applicazione di LDAP
- RFC 1959: formato del URL di LDAP
- RFC 1960: Una rappresentazione della stringa dei filtri di ricerca di LDAP
- RFC 2251: Directory Access Protocol Leggero (v3)
- RFC 2307: LDAP come servizio d'informazione della rete