Samba e OpenLDAP

Fulvio Ferroni *fulvioferroni@teletu.it*

2011.08.14

Dall'edizione 2002.09.08, il testo è stato revisionato nella sintassi e nello stile da Daniele Giacomini, *daniele ad swlibero.org*.

Indice generale

Premessa

Questo documento spiega come utilizzare gli strumenti software genericamente compresi sotto il nome Samba per far coesistere, in reti eterogenee, macchine clienti e serventi GNU/Linux con macchine clienti e serventi MS-Windows, permettendo la condivisione delle risorse e di alcuni servizi.

Dall'edizione 2006.08.18 è stato aggiunto il capitolo riguardante OpenLDAP e il titolo dell'opera è cambiato da «Samba» in «Samba e OpenLDAP».

Vengono omesse informazioni preliminari circa la configurazione di una rete locale in ambiente GNU/Linux e in ambiente MS-Windows; inoltre non vengono approfonditi i temi relativi alla configurazione dei serventi di rete MS-Windows 2000/NT. Informazioni al riguardo possono essere reperite nella vastissima letteratura disponibile su tali argomenti.
Allo stesso modo non vengono fornite istruzioni circa l'installazione di GNU/Linux e la gestione elementare del sistema. Anche in questo caso si può ricorrere a numerose fonti alternative a cominciare dagli «Appunti di Informatica Libera» di Daniele Giacomini, reperibili anche all'indirizzo ⟨*http://a2.swlibero.org*⟩

Il livello di approfondimento, nell'illustrazione dei vari temi, è volutamente non elevato: non vengono prese in esame tutte le possibili opzioni dei vari comandi o tutte le possibili direttive di un file di configurazione; ci si concentra piuttosto sulle funzionalità più importanti e le opzioni di uso più comune e frequente. Ovviamente la scelta di quali elementi trattare e di quali tralasciare è fatta dall'autore sulla base della propria esperienza di utilizzo casalingo e scolastico di Samba. Sono comunque ben accetti commenti, suggerimenti, critiche e proposte di approfondimento che possono essere inseriti in questi appunti se non violano i termini della licenza.

Non vengono presi in esame, se non in modo marginale, gli strumenti grafici per la configurazione e l'uso di Samba e OpenLDAP che, seppure molto funzionali e comodi da usare, si basano comunque sugli strumenti «testuali» standard (file di configurazione, comandi ecc.) con l'inconveniente non trascurabile di «nascondere» in certi casi i principi e le logiche di funzionamento del pacchetto. Una volta acquisita sufficiente conoscenza e padronanza sull'uso degli strumenti di base di Samba, l'utente può usare anche interfacce grafiche, riuscendo a sfruttarne in modo migliore le potenzialità e avvantaggiandosi realmente della maggiore facilità d'uso.

Dall'edizione 2005.08.18 queste dispense fanno riferimento alla versione 3 di Samba; rispetto alle edizioni precedenti è stato eliminato il capitolo riguardante Samba-TNG in quanto la principale funzionalità che quest'ultimo aveva in aggiunta rispetto alla versione standard di Samba e cioè la creazione di relazioni di fiducia con serventi MS-Windows NT, è ora inclusa nel filone principale di sviluppo del pacchetto.
Dall'edizione 2010.01.08 il capitolo relativo all'uso di Winbind fa riferimento all'accreditamento di utenti di una *active directory* e non più di un dominio «NT4».
Chi fosse ancora interessato a Samba-TNG o comunque alle edizioni precedenti di queste dispense può richiederle all'autore.

I contenuti sono stati raccolti dalla documentazione standard, da articoli di varie riviste, dall'esperienza concreta di utilizzo.

Si ringrazia il prof. Fabio Mercuri per i preziosi suggerimenti e gli aiuti riguardanti l'uso «combinato» di Samba OpenLDAP e OpenSSL.

Le prove sono state effettuate su una rete che si è evoluta nel tempo sia nelle componenti hardware che software.

Le prime versioni delle dispense erano basate principalmente sulle distribuzioni Red Hat e Fedora Core e solo marginalmente su Ubuntu; gli effetti di questa scelta sono ancora presenti in buona parte dell'opera, compresi gli esempi.

Dall'estate del 2005 si fa invece riferimento solo alle distribuzioni Debian o derivate come Ubuntu; di questa scelta risentono gli aggiornamenti del capitolo riguardante '**`Winbind`**', quelli relativi alla versione 3 di Samba e il nuovo capitolo riguardante OpenLDAP e l'uso di Samba con OpenLDAP.

Tutti i comandi e le operazioni illustrate nel proseguo di questi appunti si intendono eseguiti, in ambiente GNU/Linux, come utente '**`root`**', a meno che non venga espressamente indicata l'opportunità o l'esigenza di esecuzione come utente comune.

L'autore

Fulvio Ferroni
via Longarone,6 31030 Casier (TV)
fulvioferroni@teletu.it

Laureato in matematica, insegnante di informatica dal 1992 dopo una precedente esperienza di sistemista presso la Unisys Italia.
Usa GNU/Linux dal 1997 quando installò con successo una Slackware su un 386 usando 50 dischetti.

Fulvio Ferroni ... nel 1961.

Informazioni generali su Samba

Samba [1] è un insieme di strumenti realizzati da Andrew Tridgell per le piattaforme Unix e GNU/Linux, distribuito sotto licenza GNU GPL, curato da un piccolo gruppo di persone di varie parti del mondo coordinate dallo stesso Tridgell.

1.1 Natura di Samba

Samba utilizza il protocollo SMB (*Server Message Block*) definito per reti MS-Windows e a sua volta basato sull'interfaccia di rete NetBIOS (*Network Basic Input Output System*). SMB è stato progettato originariamente per reti molto piccole. Per permettere la connessione a reti più estese ed eterogenee, Microsoft ha sviluppato il sistema CIFS (*Common Internet File System*) ancora basato su NetBIOS.

Samba può essere a tutti gli effetti considerato una versione libera e gratuita di CIFS; con esso, una macchina GNU/Linux, può accedere alle risorse condivise di un elaboratore MS-Windows ma anche mettere a disposizione proprie risorse a clienti MS-Windows o GNU/Linux.

Più in dettaglio ecco quali sono i servizi offerti da Samba:

- servente per offrire la condivisione di file system e stampanti;
- cliente per l'accesso a risorse NetBIOS su macchine remote Unix, MS-Windows, Novell;
- *master browser* sia locale che di dominio;
- servente WINS *(Windows Internet Name Service)*;
- servente per l'autenticazione di clienti di un dominio MS-Windows;
- servente DFS *(Distributed File System)*.

Dalla versione 3 (rilasciata nel settembre 2003), abbiamo anche:

- cliente o servente di *active directory* con supporto a LDAP (*Lightweight Directory Access Protocol*) e *Kerberos5* ma senza la possibilità di sostituire un *domain controller* di *active directory* MS-Windows 2000 o 2003;
- instaurazione di relazioni di fiducia con serventi MS-Windows NT;
- strumenti per la migrazione rapida e automatica di un dominio gestito con PDC (*Primary Domain Controller*) e BDC (*Backup Domain Controller*) MS-Windows NT in un dominio gestito con PDC e BDC Samba;
- nuovo comando *net* analogo a quello presente su piattaforma MS-Windows

La versione 3 di Samba ha portato poi altre novità e miglioramenti, forse meno vistosi, ma ugualmente importanti:

- supporto alla codifica *Unicode*;
- maggiore efficienza del programma '**`Winbind`**' (gestito ora con un doppio demone[2]);
- maggiore modularizzazione di molti componenti, come la gestione utenti e l'autenticazione;

- migliore supporto per la stampa da MS-Windows 2000;
- possibilità di associare moduli VFS (*Virtual File System*) multipli ad una stessa condivisione;

È sicuramente anche il caso di elencare ciò che Samba non può fare (almeno per il momento):

- gestione mista di un dominio MS-Windows NT (cioè con PDC NT e BDC Samba o viceversa);
- sostituzione di un servente MS-Windows 2000 o 2003 per il controllo di *active directory*.

Riguardo ai protocolli coinvolti nel funzionamento di Samba, è necessario far presente che MS-Windows può incapsulare messaggi SMB sui protocolli IPX/SPX, NetBEUI e TCP/IP mentre Samba può dialogare con macchine MS-Windows solo attraverso il TCP/IP. Questa non è comunque una grande limitazione vista la diffusione ormai universale di tale protocollo

La descrizione di come NetBIOS debba operare all'interno di una rete TCP/UDP è contenuta nei documenti RFC 1001/1002. Lo standard descritto in questi documenti è noto come NBT (*NetBIOS Over TCP/IP)*) ed è alla base del funzionamento sia delle reti NetBIOS che di Samba.

1.2 Componenti di Samba

Come detto viene presa in esame la versione 3 di Samba.

I pacchetti da installare si differenziano in base alla distribuzione GNU/Linux usata; qui si fa riferimento alla Fedora Core e alla Ubuntu (derivata da Debian, quindi le informazioni fornite sono valide quasi interamente anche per quest'ultima).

Nel nome dei pacchetti si fa riferimento al generico numero di versione con la notazione ***x.y.z***; per Ubuntu, prima dell'estensione «.deb» è presente l'indicazione della versione di Ubuntu (ad esempio '**`ubuntu3`**') che qui non viene riportata.

Per la Fedora i pacchetti da usare sono:

- '`samba-`***x.y.z***`.rpm`'
- '`samba-client-`***x.y.z***`.rpm`'
- '`samba-common-`***x.y.z***`.rpm`'

Il primo pacchetto contiene gli strumenti lato servente di Samba, il secondo gli strumenti lato cliente, il terzo una serie di file indispensabili per il funzionamento sia del servente che del cliente.

Entrando più nel dettaglio:

- nel pacchetto '**`samba`**' si trovano i demoni '**`nmbd`**' e '**`smbd`**' che sono alla base del funzionamento di un servente Samba e alcuni programmi di servizio come '**`smbstatus`**';
- nel pacchetto '**`samba-client`**' sono contenuti altri programmi di servizio come '**`nmblookup`**', '**`smbclient`**', '**`smbmount`**', '**`smbumount`**', '**`smbtar`**', '**`findsmb`**';

- nel pacchetto '**samba-common**' si trovano tra gli altri i programmi '**smbpasswd**', '**net**', '**testparm**', '**winbind**' (e relative librerie), e il file di configurazione di Samba '`/etc/samba/smb.conf`' preparato con alcune impostazioni predefinite.

Per la distribuzione Ubuntu la situazione è leggermente diversa; i pacchetti da considerare sono:

- '`samba-`***x.y.z***`.deb`'
- '`samba-common-`***x.y.z***`.deb`'
- '`smbclient-`***x.y.z***`.deb`'
- '`smbfs-`***x.y.z***`.deb`'
- '`libsmbclient-`***x.y.z***`.deb`'
- '`winbind-`***x.y.z***`.deb`'

I primi due pacchetti sono più o meno analoghi a quelli di Fedora; la parte cliente viene invece suddivisa nei pacchetti '**smbfs**' (contenente i comandi '**smbmount**', '**smbumount**'), '**smbclient**' e '**libsmbclient**' (per il comando '**smbclient**'), mentre il programma '**winbind**' con le sue librerie è in un pacchetto a parte.

Tutti questi elementi sono presi in considerazione nel seguito di questo documento.

[1] **Samba** GNU GPL

[2] Un demone è un programma di servizio costantemente in esecuzione e pronto a rispondere a richieste a esso rivolte da altri processi in esecuzione localmente o da una posizione remota

Capitolo 2

Servente Samba

Fra gli strumenti che il pacchetto Samba mette a disposizione, i più importanti sono senza dubbio quelli che permettono di configurare un servente Samba.

2.1 I demoni del servente

Un servente Samba si basa su due demoni:

- '**smbd**' che fornisce i servizi di condivisione di file stampanti per i clienti SMB (che possono essere macchine MS-Windows o altre macchine GNU/Linux) e si occupa della gestione delle sessioni di comunicazione e delle autenticazioni necessarie all'accesso alle risorse che vengono offerte in condivisione dal servente; il demone avvia una copia di se stesso per ogni richiesta di servizio da soddisfare;
- '**nmbd**' che gestisce la distribuzione dell'elenco delle risorse condivise alle altre macchine della rete; può mantenere la lista delle risorse condivise (scansione della rete) e, eventualmente, risolvere i nomi NetBIOS dei vari nodi (servente WINS);

2.2 Attivazione del servente Samba

Entrambi i demoni '**smbd**' e '**nmbd**' possono essere attivati in modo autonomo, o gestiti da un supervisore dei servizi di rete (come xinetd); qui viene presa in esame solo la prima alternativa che è di gran lunga la più praticata e anche quella predefinita in molte distribuzioni GNU/Linux.

In quasi tutte le distribuzioni si trovano infatti degli script preconfezionati per l'attivazione e la disattivazione di determinati servizi; nel caso della Fedora si attivano entrambi i demoni con il comando:

```
# /etc/rc.d/init.d/smb start
```

e si disattivano con il comando:

```
# /etc/rc.d/init.d/smb stop
```

ci sono poi anche i comandi:

```
# /etc/rc.d/init.d/smb restart
```

e

```
# /etc/rc.d/init.d/smb status
```

il cui significato dovrebbe essere ovvio.

Con Ubuntu invece si usano:

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

```
# /etc/init.d/samba stop
```

```
# /etc/init.d/samba restart
```

2.3 Configurazione di un servente Samba

La configurazione di un servente Samba si basa su di un file di testo che, almeno nelle distribuzioni più diffuse, è '`/etc/samba/smb.conf`'.

Solitamente viene fornito preconfezionato e commentato, con una configurazione di base già pronta all'uso; ovviamente è possibile modificarlo per adattare il comportamento del servente alle proprie esigenze.

Prima di esaminare la struttura del file e i parametri principali di configurazione è opportuno sottolineare alcuni importanti aspetti generali:

- È indifferente usare maiuscole o minuscole a meno che tale uso non vada a interferire con le regole del sistema operativo sottostante. Se ad esempio si indica il percorso di una directory condivisa su una macchina GNU/Linux con l'opzione (che verrà descritta più avanti) '**`PATH=/USR/LOCAL`**', Samba non ha alcun problema ad accettare la direttiva ma al momento di collegarsi alla risorsa fallisce in quanto in GNU/Linux quella directory al 99 % non esiste, mentre esiste '`/usr/local/`'. È quindi consigliabile l'uso delle minuscole.
- Le righe di commento iniziano con i simboli '**`;`**' oppure '**`#`**'.
- Il carattere di continuazione riga è '**`\`**'.
- Alcune direttive di configurazione di Samba, per ragioni di compatibilità, sono ridondanti; per questo motivo uno stesso risultato si può ottenere in modi diversi.
- Per rendere effettive le variazioni fatte al file di configurazione non è necessario riavviare i demoni di Samba in quanto il file viene riletto automaticamente ogni 60 s; se si vuole forzare la rilettura basta impartire il comando: '**`kill -SIGHUP` *`n`***' dove ***n*** è il numero del processo corrispondente al demone '**`smbd`**' in funzione (per individuarlo si può eseguire '**`ps afx | grep smbd`**'). Occorre comunque notare che non tutti i cambiamenti alla configurazioni vengono necessariamente attuati subito; in particolare le variazioni della configurazione di risorse condivise rimangono congelate finché c'è qualche utente connesso a tali risorse.

Il file di configurazione è suddiviso in sezioni i cui nomi sono racchiusi tra parentesi quadrate.

Ogni sezione corrisponde a una risorsa condivisa a eccezione della sezione '**`global`**' usata per le configurazioni globali. Altre sezioni con un ruolo un po' particolare sono '**`homes`**' e '**`printers`**'.

Sezione [global]

In essa si impostano le informazioni che condizionano tutto il sistema ed eventualmente quei parametri che se non specificati vengono assunti in modo predefinito, ad esempio il nome del gruppo di lavoro (*workgroup*).

Sezione [homes]

In essa si regolano i parametri di configurazione delle directory personali degli utenti che si collegano al servente Samba.

Sezione [printers]

Consente di impostare le caratteristiche della condivisione di tutte le stampanti installate nella macchina GNU/Linux senza dover definire una condivisione separata per ognuna di esse.

All'interno del file di configurazione è possibile usare alcune variabili il cui nome viene sostituito dal rispettivo valore quando il file di configurazione viene utilizzato dai demoni '**smbd**' e '**nmbd**'.

Segue una lista delle variabili più importanti con una breve descrizione:

Variabile	Descrizione
%S	nome della condivisione corrente ('**[tmp]**', '**[homes]**', ecc.);
%P	directory principale del servizio corrente;
%u	nome dell'utente (GNU/Linux) del servizio corrente;
%g	nome del gruppo primario di '**%u**';
%U	nominativo-utente della sessione;
%D	nome del dominio MS-Windows in cui il servente Samba si integra;
%G	nome del gruppo primario di '**%U**';
%H	directory personale assegnata a '**%u**';
%v	versione in uso di Samba;
%h	nome del nodo che ha avviato il servizio Samba;
%m	nome NetBIOS della macchina cliente;
%L	nome NetBIOS assegnato al servente;
%M	nome Internet della macchina cliente;
%N	nome della directory personale ottenuta dal servizio NIS del servente;
%p	percorso della directory personale ottenuto dal servizio NIS;
%d	numero identificativo del processo corrente;
%a	architettura software della macchina remota (Samba, MS-Windows 95/98/Me/NT), diversamente sarà assegnata la parola '**UNKNOWN**';
%I	indirizzo IP della macchina cliente;
%T	data e ora corrente.

2.3.1 smb.conf: sezione global

È la sezione che appare in tutte le configurazioni di Samba, anche se non è obbligatoria. Le opzioni in essa contenute vengono applicate a tutte le altre sezioni.

Viene mostrato un esempio comprendente alcune direttive di uso comune suddivise in blocchi in base alla funzione svolta e intervallate da brevi descrizioni del loro significato:

```
[global]
#
# identificazione del servente
#
    workgroup = INF
    netbios name = pippo
    server string = Samba Server
```

La voce più importante è '**workgroup**' che assegna a Samba il dominio o il gruppo di appartenenza. Occorre assegnarla correttamente, pena conflitti nella rete paritetica (*peer-to-peer*).

La voce '**netbios name**' è di utilizzo meno frequente e serve ad assegnare al servente Samba un nome NetBIOS a piacere. Il nome NetBIOS viene infatti assegnato uguale a quello ottenuto dal DNS. Ad esempio se il nome DNS del servente fosse *muscolis.inf.besta* il nome NetBIOS sarebbe '**muscolis**'. Uno dei casi in cui è utile poter impostare un nome NetBIOS diverso da quello predefinito è quello in cui la rete è suddivisa in due o più domini DNS diversi; in questo caso potrebbe infatti anche esistere una macchina con nome *muscolis.mat.besta* che verrebbe quindi ad avere lo stesso nome NetBIOS. Ovviamente il valore di '**netbios name**' deve essere assegnato seguendo le regole dei nomi NetBIOS (unica stringa senza punti contenente i simboli alfabetici maiuscoli e minuscoli le cifre e i simboli '**!**', '**@**', '**#**', '**$**', '**%**', '**^**', '**&**', '**(**', '**)**', '**-**', '**_**' e '**~**').

Con '**server string**' si assegna semplicemente la descrizione dell'elaboratore servente.

```
#
# opzioni di rete
#
    hosts allow = 192.168.1. localhost
    hosts deny  = 172.16.244.254
    interfaces  = 192.168.1.1/24 172.16.244.1/16
    bind interfaces only = yes
```

Le direttive '**Host allow**' e '**host deny**' servono rispettivamente a specificare quali nodi possono e non possono accedere alle risorse condivise dal servente Samba. L'indicazione può essere fatta tramite il nome del nodo, il nome di dominio, il numero IP, il numero della sottorete. Nell'esempio viene concesso l'accesso a tutte le macchine della sottorete 192.168.1.$*$ e al *localhost* (è opportuno che l'accesso a *localhost* sia sempre concesso pena possibili malfunzionamenti della scansione delle risorse del servente) e viene negato alla macchina con indirizzo 172.16.244.254. Possono anche essere usate le parole chiave '**ALL**', per designare qualsiasi elaboratore, e '**EXCEPT**' per indicare un'eccezione a una regola (ad esempio '**host allow 192.168.1. EXCEPT 192.168.1.3**'). Si deve inoltre notare che in caso di assenza delle direttive '**host allow**' e '**host deny**', l'accesso è concesso a tutti in modo predefinito. Infine si tenga presente che tali direttive possono essere inserite anche in specifiche condivisioni ma con grado di priorità inferiore rispetto a quanto specificato nella sezione '**global**'.

La direttiva '**interfaces**' è utile in caso il servente Samba risieda in più di una sottorete. Se sull'elaboratore sono presenti più interfacce di rete, in modo predefinito, Samba si mette in ascolto di richieste provenienti dagli indirizzi di rete corrispondenti alla rete della prima interfaccia che trova (di solito eth0). Per fare in modo che invece risponda alle richieste provenienti da più sottoreti si deve impostare questa opzione. Nell'esempio Samba si pone in ascolto dalle sottoreti '**192.168.1.**$*$' e '**172.16.**$*$**.**$*$' (si può usare anche una notazione con maschera di rete: '**192.168.1.1/255.255.255.0**' '**172.16.244.1/255.255.0.0**').

Ponendo '**bind interfaces only = yes**' (l'alternativa è ovviamente '**no**' oppure si può

evitare di inserire questa opzione), si forza il servente a rispondere soltanto alle sottoreti corrispondenti alle interfacce indicate in '**interfaces**'. In tal caso si deve inserire tra le interfacce anche 127.0.0.1 per permettere al programma '**smbpasswd**' di potersi collegare al *localhost* e funzionare correttamente.

```
#
# opzioni per la stampa
#
    printing = cups
    printcap name = /etc/printcap
    load printers = yes
    show add printer wizard = yes
```

La direttiva '**printing**' permette di specificare il sistema di stampa in uso nel servente; per l'elenco completo delle scelte possibili si rimanda alla consultazione del manuale in linea di '**smb.conf**'.

I valori più frequentemente usati sono:

- '**bsd**', se sul servente è in funzione il sistema classico di stampa delle piattaforme GNU/Linux-Unix
- '**cups**' se è presente il più moderno sistema di stampa omonimo.

Le successive due direttive permettono di caricare automaticamente tutte le stampanti configurate nel sistema senza descriverle singolarmente, indicando a tale scopo il percorso del file 'printcap' contenente la definizione di tali stampanti. La loro configurazione relativamente a Samba viene poi indicata nell'apposita sezione '**printers**' descritta più avanti.

L'ultima opzione permette di far comparire l'icona per l'aggiunta di stampanti nella lista di condivisioni del servente Samba quando si naviga in '**risorse di rete**' da un cliente MS-Windows.

```
#
# opzioni per il log
#
    log file = /var/log/samba/%m.log
    max log size = 100
    log level = 3
```

La direttiva '**log file**' permette di indicare il file delle registrazioni per gli eventi Samba; tale file può essere unico oppure, come nell'esempio, diverso per ogni cliente che si collega al servente (il nome del file sarà '***nome_host*.log'). Altra possibilità è quella di avere un file di registrazioni per ogni utente usando opportunamente la variabile '**%U**' o '**%u**'.

Con '**max log size**' si specifica la grandezza in kbyte del file delle registrazioni, raggiunta la quale il file stesso viene rinominato con estensione '.old' e reinizializzato; il file '.old' eventualmente già esistente viene cancellato. Il valore predefinito di questo parametro è 5000; il valore zero significa nessun limite di ampiezza (scelta non consigliabile per evitare una crescita abnorme del file o dei file delle registrazioni).

La direttiva '**log level**' indica il livello di dettaglio dei messaggi annotati nel registro; il valore predefinito è zero e corrisponde a nessun messaggio. Aumentando questo valore si hanno messaggi sempre più dettagliati; è comunque sconsigliabile un livello superiore a 3.

```
#
# opzioni per l'accesso alle condivisioni
#
    encrypt password = yes
    null password = yes
    guest account = utentesmb
#   restrict anonymous = 2
    security = share
# in alternativa
;  security = user
;  security = server
;  security = domain
# altri parametri nel caso di security = user
;  smb passwd file = /etc/samba/smbpasswd
;  username map = /etc/samba/smbusers
# altri parametri nel caso di security = server o domain
;  password server = SERVER_NT
```

La direttiva '**`encrypt password = yes`**' è praticamente obbligatoria se il servente Samba deve «convivere» con macchine equipaggiate con sistemi operativi MS-Windows 98/NT o più recenti che usano le parole d'ordine cifrate.

La direttiva '**`null password = yes`**' permette di avere utenti Samba con parola d'ordine nulla (il valore predefinito è '**`no`**'.)

La direttiva '**`guest account`**' indica il nome di un utente generico al quale può essere consentito l'accesso alle condivisioni (da usare nel caso di '**`security = share`**', come dettagliato più avanti). Tale utente deve essere definito nel sistema GNU/Linux senza directory personale e senza shell, aggiungendo al file '`/etc/passwd`' la riga:

```
utentesmb::499:499:utente generico samba:/dev/null:/dev/null.
```

Il valore predefinito è '**`nobody`**'.

La direttiva '**`restrict anonymous = 2`**', se attivata permette di evitare i collegamenti anonimi al servente Samba.

Il parametro '**`security`**' è di importanza fondamentale e richiede una trattazione leggermente più ampia.

In ambito SMB/CIFS esistono due tipi di controllo di accesso: «a livello di condivisione» o «share level» e «a livello utente» o «user level»; con Samba si ha una maggiore flessibilità rispetto ai serventi MS-Windows in quanto è presente il livello «share» e ci sono ben quattro varianti del livello «user»:

- «user»
- «domain»
- «server»
- «ADS»

2.3.1.1 Livello di sicurezza «share»

Con l'impostazione '**security = share**' si ha il controllo di accesso a livello di condivisione: il cliente che vuole accedere a una risorsa invia ogni volta una parola d'ordine e nessun nominativo-utente. Samba tenta di dedurre il nominativo-utente dalla direttiva '**valid users**' eventualmente inserita nella sezione di condivisione di quella risorsa (come illustrato in seguito), oppure dal nome dell'elaboratore cliente, o, in caso di insuccesso, da quanto indicato con il parametro '**guest account**' (questo solo se fra i parametri di condivisione è indicato '**guest ok = yes**' e '**guest only = yes**').

Questo livello di sicurezza si usa, soprattutto nel caso di utenti MS-Windows e GNU/Linux non coincidenti, per condividere porzioni di file system quando si ha interesse a far sì che tutti gli utenti abbiano gli stessi diritti sui file condivisi (con l'impostazione '**guest account = utentesmb**' il «proprietario» delle risorse condivise sarà sempre l'utente GNU/Linux '**utentesmb**' qualunque sia il nominativo dell'utente MS-Windows che si connette).

2.3.1.2 Livello di sicurezza «user»

Con l'impostazione '**security = user**', che è quella predefinita, si ha il controllo di accesso a livello di utente: il cliente che vuole accedere a una risorsa invia al momento della connessione una coppia utente-parola d'ordine in base alla quale avviene l'autenticazione da parte del servente. Se la connessione viene accettata il cliente può accedere a tutte le risorse condivise senza doversi autenticare nuovamente a ogni accesso.

Tutto questo è possibile grazie all'esistenza di un sistema di controllo detto SAM (*System Account Manager*) basato su un archivio utenti che può essere gestito in vari modi:

- con file di testo '`/etc/samba/smbpasswd`'; è il modo più tradizionale e anche più semplice ma sicuramente poco versatile;
- con archivi TBDSAM (*Tiny Database SAM*) che è la scelta predefinita con Samba 3;
- con serventi locali o remoti NIS, LDAP, MySQL o altri ancora.

Le prime due soluzioni, che sono quelle considerate nella prima parte di queste dispense, sono consigliabili in presenza di domini con un numero limitato di utenti, non più di qualche centinaio; per domini più consistenti, che solitamente richiedono la presenza di più PDC, in considerazione del fatto che Samba non prevede la replica diretta degli archivi utente, è più opportuno ricorrere a soluzioni basate su LDAP.

La scelta del tipo di archivio utenti da utilizzare si effettua nel seguente modo:

```
#
# tipo di archivio utenti
#
    passdb backend = smbpasswd
# oppure
;   passdb backend = tdbsam
```

Ovviamente con la prima scelta si attiva l'archivio su file di testo, con l'altra l'archivio TBDSAM.

Gli utenti Samba vengono aggiunti con il comando '**smbpasswd**' illustrato in un paragrafo successivo (2.4).

Valide alternative, specie se non si usa il semplice archivio su file di testo, sono costituite da:

- comando '**pdbedit**', che permette anche di travasare utenti da un tipo di archivio ad un altro;
- comando '**net**';
- comando, via rete, '**usrmgr.exe**' di MS-Windows.

L'uso di questi comandi non viene qui approfondito, solo il comando '**net**' viene ripreso in paragrafi successivi (2.3.1.4) e (2.4).

È di fondamentale importanza notare che comunque il nome utente Samba deve coincidere con il nome di un utente definito nel sistema GNU/Linux, in quanto quest'ultimo deve essere sempre in grado di assegnare un proprietario «valido» agli eventuali file creati o copiati dall'utente connesso all'interno della risorsa condivisa.

Se ciò costituisse un problema si può ricorrere all'uso di un file di corrispondenza tra utenti GNU/Linux e utenti MS-Windows, il cui nome è indicato con la direttiva '**username map =**'. Un valore abbastanza comune di tale parametro è '`/etc/samba/smbusers`'. Tale file conterrà delle righe così formate:

```
nome_linux = nome_smb_1 nome_smb_2...
```

Per ovvi motivi di sicurezza, entrambi i file '`smbusers`' e '`smbpasswd`', devono essere accessibili sia in scrittura che in lettura dal solo utente '**root**'.

2.3.1.3 Livello di sicurezza «server»

Con l'impostazione '**security = server**' si ha lo stesso controllo di accesso visto nel caso di '**user**' con la differenza che l'utenza viene controllata su un servente esterno (solitamente un PDC MS-Windows, ma può essere anche un PDC Samba) il cui nome (NetBIOS) viene indicato con la direttiva '**password server =**'.

2.3.1.4 Livello di sicurezza «domain»

Con l'impostazione '**security = domain**' si ha ancora lo stesso controllo di accesso visto nel caso di '**user**', ma questa volta il servente Samba va a inserirsi in un dominio MS-Windows NT/2000.

Per ottenere questo risultato occorre fermare i demoni di Samba, quindi aggiungere il servente Samba al dominio NT sul PDC usando il *server manager* di MS-Windows NT ». A questo punto si deve eseguire il comando:

```
# net rpc join -Unome_utente%parola_d'ordine
```

oppure (con Samba 2.x):

```
# smbpasswd -j nome_dominio -r nome_serv -Unome_utente%parola_d'ordine
```

Il ***nome_dominio*** deve essere quello a cui si vuole unire il servente Samba e lo stesso indicato nella direttiva '**workgroup**' di '`smb.conf`'; il ***nome_serv*** deve coincidere con il valore di '**password server**'; ***nome_utente*** e relativa ***parola_d'ordine*** devono rappresentare un'utenza con privilegi sufficienti ad aggiungere un'utenza nuova nella macchina MS-Windows NT/2000; ***nome_dominio*** e ***nome_serv*** nel comando '**net**' non servono in quanto vengono letti automaticamente in '`smb.conf`'.

Ultimate queste operazioni occorre naturalmente riavviare il servizio Samba.

In questo modo la macchina con Samba viene ad essere un *member server* del dominio, cioè una stazione che non contiene copie dell'archivio utenti del dominio stesso, ma offre comunque risorse condivise.

Il vantaggio principale dell'impostazione '**domain**' rispetto a quella '**server**' consiste nel fatto che il PDC risulta meno carico, in quanto non è più necessaria una connessione di rete permanente tra esso e il servente Samba. Quest'ultimo infatti effettua una chiamata RPC (*Remote Procedure Call*) solo al momento dell'autenticazione e non necessita di essere costantemente connesso al PDC come avviene nel caso del livello di sicurezza '**server**'.

Per questo, e per altre questioni legate alla sicurezza, è sconsigliato usare il livello '**server**' a vantaggio del livello '**domain**'.

Per concludere occorre notare che anche con questa impostazione (come pure per quella '**server**') è necessario tenere allineati gli elenchi degli utenti dal lato MS-Windows e dal lato GNU/Linux (il motivo è stato illustrato nel paragrafo del livello '**user**'). Ci sono però due direttive della sezione '**global**' che permettono di automatizzare l'aggiornamento degli utenti GNU/Linux:

```
add user script = script_1 %u
delete user script = script_2 %u
```

La prima entra in azione quando, a seguito di una connessione di un cliente MS-Windows, Samba si rivolge al servente di dominio per l'autenticazione con esito positivo ma l'utente GNU/Linux corrispondente non esiste; ovviamente lo script ***script_1*** deve essere scritto in modo adeguato affinché crei l'utente ricevendolo come parametro dalla variabile '**%u**'.

In modo speculare si usa l'altra direttiva che entra in azione quando, a seguito di una connessione di un cliente MS-Windows, Samba si rivolge al servente di dominio per l'autenticazione con esito negativo ma il corrispondente utente GNU/Linux esiste; in questo caso ***script_2*** deve provvedere a cancellare l'utente in questione.

Un modo molto più elegante di risolvere il problema consiste nel ricorrere ad un archivio utenti basato su LDAP contenente le informazioni di accreditamento sia per Samba che per il sistema GNU/Linux (utilizzando a tale scopo l'interfaccia '**ldapsam**' fra l'archivio utenti e Samba e i moduli '**nsswitch**' e '**PAM**' per l'autenticazione su GNU/Linux); tali argomenti sono trattati nel capitolo 9.

2.3.1.5 Livello di sicurezza «ADS»

Con l'impostazione '**security = ADS**' dove ADS significa *Active Directory Server*, disponibile a partire da Samba 3, la stazione con Samba diviene un *member server* di una *active directory* di MS-Windows 200x, utilizzando l'autenticazione Kerberos5 e LDAP per l'archivio utenti.

Maggiori dettagli a questo proposito vengono forniti nel capitolo sull'uso di '**Winbind**' (7.2).

2.3.2 smb.conf: sezioni generiche di condivisione

Una sezione indicante una risorsa condivisa può avere un nome a piacere purché sempre racchiuso tra parentesi quadrate.

Vengono adesso illustrati alcuni esempi allo scopo di descrivere almeno le direttive più importanti:

```
#
# condivisione 1: una directory accessibile solo a certi utenti
#
[Pagine WWW]
   comment = Dir per le pagine web
   browseable = yes
   public = no
   path = /var/www
   writable = yes
   valid users = utente1 utente2
```

La direttiva '**comment**' serve ad associare una descrizione alla risorsa condivisa.

L'opzione '**browseable**' permette di rendere visibile o no la risorsa agli utenti che si connettono al servente.

'**public**' è un sinonimo di '**guest ok**'; in questo esempio non si vuole che la risorsa sia accessibile per l'utente generico.

'**path**' permette di indicare il percorso della risorsa sul sistema GNU/Linux.

'**writable**' serve a concedere o negare l'accesso in scrittura (un sinonimo è '**writeable**').

'**valid users**' indica quali sono gli utenti che possono accedere alla risorsa. È anche possibile indicare gruppi di utenti GNU/Linux con la sintassi '***+nome_gruppo***' e gruppi di utenti NIS (ovviamente deve essere presente in rete un servente NIS) con la sintassi '***&nome_gruppo***' e anche entrambi con la sintassi '***+&nome_gruppo***' o '***&+nome_gruppo***', o ancora '***@nome_gruppo***'.

A proposito della opzione '**valid users**' è importante sottolineare come in Samba ci siano due livelli di controllo di accesso alle risorse condivise:

- il primo, e più importante, è quello che si ottiene con l'impostazione del livello di sicurezza e in pratica stabilisce chi ha diritto di collegarsi al Servente;
- il secondo, molto più granulare, viene effettuato sulle singole risorse grazie ad opzioni come '**valid users**', '**public**' '**write list**' ed altre ancora, grazie alle quali si può stabilire quali utenti possono accedere ad ogni singola risorsa.

```
#
# condivisione 2: una directory pubblica = accessibile a tutti
#
[public]
   comment = Dir pubblica
   browseable = yes
   guest ok = yes
   path = /usr/local/public
   writable = yes
```

```
#
# condivisione 3: una directory pubblica = accessibile a tutti in cui tutti
# gli utenti possano creare, modificare, cancellare tutti i file
#
[temp]
   comment = Dir pubblica plus
   browseable = yes
   guest ok = yes
   guest only = yes
   path = /tmp
   writable = yes
```

La differenza tra le due condivisioni è molto sottile ma anche interessante. La presenza di '**guest ok = yes**' permette le connessioni anonime; eventuali file creati o copiati nella directory sarebbero di proprietà dell'utente indicato in '**guest account**' in caso di accesso anonimo, oppure dell'utente effettivo in caso esso fosse registrato in GNU/Linux. Se è presente anche '**guest only = yes**' invece il proprietario è sempre l'utente fittizio '**ospite**' anche nel caso l'utente collegato fosse riconosciuto regolarmente da GNU/Linux; in questo caso quindi tutti gli utenti possono fare tutte le operazioni con qualsiasi file presente nella directory condivisa.

Un modo alternativo per ottenere lo stesso risultato è quello di usare la direttiva '**force user = *nome_utente***'; in questo modo Samba assegna lo stesso nominativo-utente a chiunque si connetta alla risorsa.

```
#
# condivisione 4: un cd-rom
#
[cd]
   comment = CD-ROM
   preexec = mount /mnt/cdrom
   postexec = umount /mnt/cdrom
   browseable = yes
   public = yes
   path = /mnt/cdrom
   writable = no
```

Il significato delle impostazioni è ovvio compreso quello delle due opzioni '**preexec**' e '**postexec**'. In caso si tema che possano connettersi utenti sprovvisti dei privilegi per montare e smontare il CD, si possono sostituire le due direttive rispettivamente con '**root preexec**' e '**root postexec**' che svolgono lo stesso compito ma con i privilegi dell'utente '**root**'.

```
#
# condivisione 5: una directory privata con permessi preimpostati
#
[privata]
   comment = Dir privata
   browseable = yes
   path = /usr/local/private
   writable = no
   public = no
   write list = pippo pluto
   create mask = 0644
   directory mask = 0644
```

In questo esempio nella directory non sarebbe possibile scrivere, ma la presenza della direttiva '**write list**' permette di impostare permessi di scrittura, per gli utenti indicati, indipendentemente da quanto specificato negli altri parametri. Si deve però notare che i permessi impostati a livello di sistema su quella risorsa hanno sempre il sopravvento su quanto specificato nel file di configurazione di Samba (in altre parole, se '/usr/local/private' è di proprietà dell'utente '**root**' e i permessi sono impostati a 600_8, gli altri utenti non possono né leggere né scrivere alcunché in quella directory condivisa).

Le ultime due direttive, infine, servono a indicare i permessi con cui verranno creati file e directory all'interno della risorsa condivisa; il valore predefinito è 0755_8

Altre direttive importanti sono:

```
admin users = tizio
follow symlinks = no
```

Con la prima si indica che l'utente '**tizio**' ha gli stessi privilegi dell'utente '**root**' sulla condivisione e quindi non risente di eventuali limitazioni dovute ai permessi sui file; con la seconda si impedisce che vengano seguiti i collegamenti simbolici evitando che chi accede alla condivisione possa accedere anche a file che si trovano all'esterno di questa.

2.3.3 smb.conf: sezione «homes»

La sezione '**homes**' viene utilizzata affinché ogni utente possa avere accesso a una propria directory personale sul servente Samba che potrà anche coincidere con la directory personale GNU/Linux di quell'utente. Un esempio di definizione può essere il seguente:

```
#
# directory personali degli utenti
#
[homes]
    comment = directory home
    browseable = no
    writable = yes
    path = usr/local/samba/%S
```

Qui è importante l'impostazione che impedisce la scansione della risorsa in modo che essa non appaia con il nome '**homes**' a tutti gli utenti.

La presenza di '**path**' serve a fare in modo che questa directory non coincida con quella GNU/Linux dell'utente (come da impostazione predefinita).

La logica di funzionamento è la seguente: quando l'utente si connette, se l'utenza è accettata, viene creata dal servente Samba una condivisione con le caratteristiche specificate nella sezione '**homes**' ma con un nome uguale a quello dell'utente connesso.

2.3.4 smb.conf: sezione «printers»

Con questa sezione si impostano i parametri di configurazione di tutte le stampanti definite nel sistema GNU/Linux a patto che nella sezione '**global**' siano state inserite le direttive seguenti:

```
load printers = yes
printcap name = /etc/printcap
```

L'alternativa, che consiste nel definire le varie stampanti come singole risorse condivise, non viene presa in esame in questa sede.

Un esempio di configurazione per le stampanti è il seguente:

```
#
# Stampanti
#
[printers]
   comment = stampanti
   path = /var/spool/samba
   browseable = no
   printable = yes
   public = yes
   writable = no
```

Qui occorre notare che il parametro '**path**' serve a impostare una directory per la coda di stampa, diversa da '`/tmp/`' che è quella predefinita.

Altra direttiva da segnalare è '**printable**' con la quale si attiva la coda di stampa.

2.4 Programmi ausiliari per un servente Samba

Un primo strumento molto utile è '**testparm**' con il quale si verifica la correttezza sintattica delle impostazioni scritte nel file '`smb.conf`'.

Il comando da eseguire è:

```
# testparm /etc/samba/smb.conf
```

si ottiene una risposta suddivisa in due parti: prima il resoconto del controllo sintattico del file di configurazione, poi l'elenco di tutte le direttive inserite in esso.

Se si usa l'opzione '**-v**' si ottiene la lista di tutti i parametri di configurazione di Samba compresi quelli non presenti in '`smb.conf`' ai quali viene assegnato il valore predefinito.

Altro programma di fondamentale importanza è '**smbpasswd**', già visto in precedenza a proposito della connessione di Samba a un dominio MS-Windows NT, ma che si usa principalmente per definire utenti e parole d'ordine relative. La sintassi in questo caso è:

```
smbpasswd [-a] [-x] [nominativo]
```

L'opzione '**-a**' permette di inserire un nuovo utente e poi di definirne la parola d'ordine; l'opzione '**-x**' permette invece di eliminarlo; se non si indica alcuna opzione si esegue solo il cambio della parola d'ordine per l'utente. Il nominativo-utente che si può inserire alla fine della riga di comando è quello sul quale il comando opera (se non viene indicato, si fa riferimento in modo predefinito all'utente GNU/Linux che esegue il comando).

Altro strumento utile per la gestione utenti e gruppi anche da remoto e per numerosi altri utilizzi è il nuovo comando '**net**'; le sue potenzialità sono notevoli e non vengono qui esaminate in dettaglio rimandando alla consultazione del manuale in linea.

Di seguito vengono riportati alcuni esempi di utilizzo riguardanti principalmente la gestione utenti:

```
# net rpc user -S nome_serv -UAdmin%psw_admin
```

```
# net rpc user add nome_utente -S nome_serv -UAdmin%psw_admin

# net rpc password nome_utente -S nome_serv -UAdmin%psw_admin

# net rpc user delete nome_utente -S nome_serv -UAdmin%psw_admin

# net rpc -l share -S nome_serv -UAdmin%psw_admin

# net rpc -l file -S nome_serv -UAdmin%psw_admin
```

I comandi servono nell'ordine a:

- elencare gli utenti;
- definire un nuovo utente;
- cambiare la parola d'ordine di un utente;
- cancellare un utente;
- elencare le condivisioni di un servente;
- elencare i file aperti in un servente.

Tutti i comandi vengono eseguiti sul servente '**nome_serv**' come utente '**Admin**' con relativa parole d'ordine e utilizzando il protocollo '**rpc**'.

A tale proposito è importante notare che i protocolli utilizzabili con il comando '**net**' sono tre:

- '**ads**' per le *active directory*;
- '**rap**' per macchine MS-Windows 9x/me/NT3;
- '**rpc**' per macchine MS-Windows NT4/200x;

Se il protocollo non viene specificato, il comando cerca di determinarlo in modo automatico.

Infine può essere molto utile anche il comando '**smbstatus**' per avere un rapporto (con l'opzione '**-d**' anche dettagliato) delle connessioni Samba attive.

Maggiori dettagli sulla configurazione di un servente Samba si possono trovare anche nella documentazione fornita insieme al pacchetto in '`/usr/share/doc/samba-`***x.y.z***' ('`/usr/share/doc/samba`' per Ubuntu), oppure consultando la pagina di manuale *smb(5)*.

Capitolo 3

Samba dal lato cliente

Da una macchina GNU/Linux è possibile connettersi a un servente Samba o a macchine MS-Windows per accedere alle risorse che queste condividono grazie a una serie di strumenti che costituiscono il «lato cliente» di Samba.

A questi strumenti può essere aggiunto il nuovo comando '**net**' utilizzabile anche per collegarsi a serventi SMB/CIFS come visto in uno degli esempi dell'ultimo paragrafo del capitolo precedente (2.4).

3.1 smbclient

Il primo programma da esaminare è '**smbclient**' che si usa fondamentalmente in due maniere:

```
smbclient -L nome_serv_samba
```

per avere la lista delle risorse condivise dal servente; oppure:

```
smbclient nome_servizio [-U nominativo_utente]
```

per connettersi alla risorsa ***nome_servizio*** (ad esempio '`//serv_samba/public`'). Se la connessione ha successo si ha a disposizione un'interfaccia testuale del tutto simile a quella del programma '**ftp**' tradizionale, dove si possono eseguire più o meno gli stessi comandi ('**get**', '**put**', '**cd**', '**pwd**', ecc.).

Un'opzione importante è '**-I**' seguita da un numero IP, con la quale si può appunto indicare il numero IP del servente a cui ci si vuole connettere.

3.2 smbmount

Sicuramente l'interfaccia messa a disposizione da '**smbclient**' non è il massimo della comodità; sarebbe molto meglio poter disporre della risorsa condivisa da un'altra macchina come se fosse una risorsa locale.

Anche questo si può ottenere con Samba, in particolare grazie a '**smbmount**', oppure grazie al più «moderno» '**mount.cifs**' (o '**mount -t cifs**'che è sostanzialmente equivalente riguardo alla sintassi e alle opzioni.

La sintassi è:

```
smbmount nome_servizio punto_di_innesto [-o opzioni]
```

Ad esempio:

```
# smbmount //serv_samba/public /mnt/dir_samba -o username=tizio%parola_d'ordine
```

Per smontare la risorsa si può usare il comando:

```
smbumount punto_di_innesto
```

È anche possibile ottenere l'inserimento automatico della risorsa all'avvio di GNU/Linux con la riga seguente nel file '`/etc/fstab`':

```
//serv_samba/public     /mnt/dirsamba     smbfs     username=tizio%parola_d'ordine
```

È importante ribadire che in questa sede non viene illustrata la sintassi completa dei vari comandi con l'elenco di tutte le opzioni possibili. Per avere queste informazioni si deve consultare la documentazione dei vari pacchetti e il manuale in linea (per esempio *smbpasswd(8)*).

3.3 Uso con Samba di stampanti MS-Windows

Per usare su una macchina GNU/Linux una stampante condivisa da una macchina MS-Windows, occorre impostare in modo opportuno il file di definizione delle stampanti GNU/Linux: '`/etc/printcap`'.

Ecco un esempio di righe di configurazione da aggiungere a tale scopo:

```
stsamba:\
   :mx=0:\
   :sh:\
   :sd=/var/spool/lpd/stsamba:\
   :lp=/dev/null:\
   :af=/var/spool/lpd/stsamba/acct:\
   :if=/usr/bin/smbprint:
```

Una breve spiegazione sul significato di queste righe è necessaria, anche se lo studio del file di configurazione delle stampanti in GNU/Linux esula dallo scopo di questo documento:

'**`mx=0`**' indica nessun limite di grandezza dei file da stampare;

'**`sd=`**' indica la directory della coda per questa stampante;

'**`lp=/dev/null`**' indica che la stampante non è collegata ad alcuna porta (non è locale);

'**`af=`**' indica il nome del file per registrare le transazioni;

'**`if=`**' indica il nome del filtro da usare per la stampa; quello usato nell'esempio è un filtro fornito in modo predefinito con la distribuzione Red Hat all'interno del pacchetto '**`samba-client`**'.

Affinché il tutto funzioni è poi necessaria la presenza del file '`.config`' nella directory della coda indicata con la riga '**`sd=`**'. Un esempio del suo contenuto è il seguente:

```
   share="//serv_samba/nome_stampante_condivisa"
   user="tizio"
   password="blablabla"
```

Data la complessità della configurazione manuale di una stampante SMB in GNU/Linux (e anche di una stampante in generale), può essere consigliabile l'uso di strumenti appositi che semplificano molto queste operazioni; un esempio può essere '**`printconf-gui`**' oppure, se si usa CUPS, la sua interfaccia WEB richiamabile collegandosi a *`localhost:631`*.

3.4 Altri strumenti utili

Altri comandi presenti nel pacchetto '**samba-client**' sono:

'**nmblookup**' che permette di trovare il numero IP di una macchina fornendo il nome NetBIOS;

'**findsmb**' che fornisce informazioni sui serventi Samba presenti in rete;

'**smbtar**' che permette di effettuare copie di sicurezza di risorse SMB su unità a nastro installate sul servente GNU/Linux.

Per i dettagli di uso di questi comandi si rimanda ai rispettivi manuali in linea.

Capitolo 4

Samba come servente WINS

In una rete basata su NetBIOS la risoluzione dei nomi è basata sull'invio di messaggi circolari; quando un elaboratore con MS-Windows si collega in rete invia un messaggio a tutti informando sul proprio nome e sul proprio indirizzo. È ovvio che un sistema di questo tipo può essere efficiente solo su piccole reti e in assenza di sottoreti multiple (i router di solito sono configurati per bloccare i messaggi circolari).

La soluzione a questo problema proposta da Microsoft consiste nell'introduzione di un servente WINS allo scopo di gestire una tabella contenente le associazioni tra nomi NetBIOS e indirizzi IP. I clienti della rete hanno impostata l'indicazione del numero IP del servente WINS in modo che, quando devono qualificarsi o chiedere informazioni circa l'identità di altre macchine, si rivolgono direttamente al servente senza generare traffico superfluo.

Un servente Samba può essere configurato sia per svolgere la funzione di servente WINS sia per essere cliente di un servente WINS già presente in rete (ovviamente non sono possibili entrambe le impostazioni contemporaneamente).

Nella sezione '**`global`**' si deve aggiungere il seguente parametro per attivare il servente WINS:

```
    wins support = yes
```

Invece per indicare a Samba qual è il servente WINS già attivo in rete si usa:

```
    wins server = 192.168.1.1
```

Capitolo 5

Samba e la scansione della rete

In una rete locale le macchine hanno sempre una lista delle altre macchine attive che si chiama lista di *browse*. Il servente che la gestisce si chiama *master browser* locale se riveste questo ruolo solo per una sottorete, se invece mantiene la lista per tutta la rete locale diviene un *master browser* di dominio.

Siccome in una rete le macchine possono essere collegate e scollegate in ogni momento, il *master browser* locale aggiorna continuamente la lista e la invia alle macchine che ne fanno richiesta. Il *master browser* di dominio invece raccoglie le liste di ogni sottorete e le mette a disposizione dei *master browser* locali.

Un elaboratore diventa *master browser* locale a seguito di una «elezione» che può essere effettuata in qualunque momento (ad esempio quando un nuovo elaboratore si presenta in rete). Riguardo a tale elezione Microsoft assegna ai suoi sistemi operativi un valore, detto livello, che cresce sempre di più per i sistemi operativi più recenti (ad esempio una macchina con MS-Windows 98 ha livello 1, una con MS-Windows NT 4 Workstation ha livello 16, una con MS-Windows NT 4-Server ha livello 32, una con MS-Windows 2000-Server ha livello 64).

L'elezione avviene sulla base di questo valore; in caso di parità viene esaminata la funzione svolta dalla macchina (senza entrare in dettagli eccessivi, basti sapere che un PDC diviene sempre anche PDM); in caso di ulteriore parità viene scelta la macchina che è da più tempo in rete.

Con Samba il livello per la partecipazione all'elezione può essere scelto in sede di configurazione. Nell'esempio seguente vengono mostrate le direttive da inserire nella sezione '**`global`**' affinché il servente Samba sia *master browser* locale prevalendo su macchine MS-Windows equipaggiate fino a NT-server:

```
local master = yes
os level = 34
preferred master = yes
```

La terza direttiva serve ad attivare il *preferred master bit* del servente Samba, in modo che il proprio servente prevalga al momento dell'elezione su macchine con uno stesso sistema operativo.

Se si vuole che il servente Samba divenga *master browser* di dominio occorre inserire anche:

```
domain master = yes
```

In caso però che il servente partecipi a un dominio NT è preferibile che il ruolo di *master browser* di dominio sia lasciato al PDC (che comunque, come viene descritto nel prossimo capitolo, può essere lo stesso servente Samba).

A proposito della scansione della rete, nel caso siano presenti sottoreti multiple, è opportuno ricordare le seguenti regole generali:

- ci deve essere una macchina MS-Windows o Samba che faccia da *master browser* locale per ciascuna sottorete (se nella sottorete c'è gia un *master browser* di dominio non è necessario anche il *master browser* locale);
- almeno una macchina MS-Windows o Samba deve essere *master browser* di dominio per il gruppo di lavoro;
- ogni *master browser* locale deve sincronizzarsi con il *master browser* di dominio.

In caso nella rete non sia presente un *master browser* di dominio e ci siano però delle sottoreti multiple, Samba mette a disposizione due direttive utili per la sincronizzazione (da inserire sempre nella sezione '**global**'):

```
remote announce = 192.168.1.255/INF  192.168.2.255/INF
remote browse sync = 192.168.3.255 192.168.4.255
```

Con '**remote announce**' il servente Samba fornisce l'elenco di scansione anche ad altre sottoreti. Se si conoscono i numeri IP dei *master browser* locali si possono indicare tali numeri, altrimenti (come nell'esempio) si indicano degli indirizzi broadcast. In pratica con i valori indicati, il servente Samba segnalerà la sua presenza a tutte le macchine delle sottoreti 192.168.1.$*$ e 192.168.2.$*$ (del gruppo di lavoro INF) e quindi anche ai relativi *master browser* locali.

L'altra direttiva ha uno scopo simile nel caso però i *master browser* locali delle sottoreti siano altri serventi Samba (anche in questo caso si possono indicare gli indirizzi precisi dei serventi o degli indirizzi broadcast). Nell'esempio, un servente Samba contatta altri serventi Samba delle sottoreti 192.168.3.$*$ e 192.168.4.$*$, con i quali sincronizza le liste di scansione.

A proposito della possibilità di usare gli indirizzi broadcast ci si deve sincerare che i router della rete non siano configurati per bloccare il traffico broadcast tra sottoreti diverse.

Capitolo 6

Autenticazione di utenti MS-Windows con Samba

A partire dalla versione 2.0 è possibile configurare Samba come *domain controller* e autenticare gli utenti degli elaboratori clienti MS-Windows 95/98 sostituendo un servente MS-Windows NT/2000.

Con la versione 2.1 è stata data la possibilità di accreditare anche clienti MS-Windows NT.

Dalla versione 2.2, Samba può accreditare anche clienti MS-Windows 2000/XP ed è stato aggiunto il demone '**winbindd**' che consente di usare un *domain controller* MS-Windows come servente per le utenze, allineando del tutto le utenze di GNU/Linux con quelle di MS-Windows e centralizzando la loro gestione su un solo sistema.

Le novità introdotte con la versione 3, unitamente alle funzionalità ancora non presenti, sono state già elencate nel capitolo introduttivo (1.1).

In questa sede viene presa in esame solo la configurazione di Samba come PDC per l'accreditamento di clienti MS-Windows 95/98/Me/NT/XP.

Di seguito viene presentato un possibile file '`smb.conf`' con le definizioni necessarie affinché Samba sia un PDC:

```
[global]
   netbios name = serv_samba
   workgroup = INF
   server string = Samba Server NT
   log file = /var/log/samba/%m.log
   max log file = 50
   security = user
   encrypt password = yes
   passdb backend = smbpasswd
   local master = yes
   preferred master = yes
   os level = 33
   domain master = yes
;
   domain logons = yes
;
#  script di accesso fisso per tutti
   logon script = logon.bat
#  oppure uno per ogni cliente
;  logon script = %m.bat
#  oppure uno per ogni utente
;  logon script = %U.bat
;
#  profili utenti
   logon path =  \\serv_samba\profile\%U


[netlogon]
   comment = Directory degli script di inizializzazione
   path = /home/netlogon
   read only = yes
   guest ok = yes
   browseable = no
```

```
[home]
   comment = Dir utente
   path = /home/%U
   browseable = yes
   writable = yes

[public]
   comment = Dir pubblica
   path = /home/public
   browseable = yes
   writable = yes
   public = yes
   create mask = 0777
```

6.1 Domain logons

È la direttiva che permette di configurare Samba come PDC in quanto lo imposta come servente di autenticazione di dominio.

6.2 Logon script

Samba consente l'esecuzione degli script di accesso di MS-Windows ('`.BAT`' o '`.CMD`'). Tali script vengono eseguiti sul cliente al momento della connessione di un utente al dominio ma sono memorizzati sul servente e vengono quindi trasferiti attraverso la rete. Ovviamente sono molto utili per impostare dinamicamente le configurazioni di rete per gli utenti quando si connettono.

L'opzione '**`logon script`**' permette appunto di indicare il nome dello script da eseguire quando l'utente si collega; come si vede dall'esempio può essere uno script unico, valido per tutti, oppure dipendente dal cliente o dal nome utente.

Sul servente GNU/Linux questi script vengono memorizzati nella directory indicata nella condivisione '**`netlogon`**', che viene descritta più avanti.

Una cosa importante da ricordare è che gli script di accesso vengono eseguiti in ambiente MS-Windows e devono quindi avere righe terminanti con i caratteri di <*CR*> e <*LF*>, invece del solo <*LF*> di un sistema GNU/Linux.
Nel caso tali script siano scritti in ambiente Linux si può utilizzare il comando '**`todos`**' (che fa parte del pacchetto '**`tofrodos`**') per convertirli al formato MS-Windows.

L'esempio seguente di script di accesso, definisce un disco di rete '`W:`' su una condivisione di Samba:

```
   echo Connette disco di rete
   net use w: \\serv_samba\dati
```

6.3 Logon path

In MS-Windows 95/98 ciascun utente può avere il proprio profilo comprendente informazioni sull'aspetto della scrivania grafica, sulle applicazioni che appaiono nel menù *Start*, sullo sfondo e altre ancora. Tale profilo può essere memorizzato direttamente su un elaboratore cliente e si chiama allora «profilo locale», oppure sul servente e si chiama «profilo di roaming», in quanto l'utente ha a disposizione sempre lo stesso ambiente anche spostandosi da un cliente all'altro.

La direttiva '**logon path**' viene usata per indicare dove vengono memorizzati i profili dei vari utenti.

6.4 Logon home e logon drive

Con '**logon home**' si indica la posizione della directory personale di un utente, che può essere diversa da quella indicata nella sezione '**homes**'.

Con '**logon drive**', da usare solo in caso di clienti MS-Windows NT, si indica la lettera del disco su un cliente in cui vengono abbinate le directory personali indicate con '**logon home**'.

6.5 Sezione (netlogon)

In questa sezione viene configurata una condivisione speciale che serve a contenere gli script di accesso. La configurazione scelta nell'esempio (sola lettura, pubblica, non visibile alla scansione delle risorse) è dettata dal ruolo particolare che svolge.

6.6 Definizione delle utenze per macchina

Nel caso nella rete siano presenti dei clienti MS-Windows NT/2000/XP Workstation/Professional, per essi devono essere creati sul PDC i cosiddetti *machine account* in aggiunta alle utenze normali. Ovviamente, tali utenze speciali devono essere inserite sia come utenti Samba che come utenti del sistema GNU/Linux che ospita il servente; i comandi necessari sono i seguenti:

```
# /usr/sbin/groupadd machines
```

Per creare un gruppo, denominato ***machines***, in cui definire le utenze macchina;

```
# /usr/sbin/useradd -d /dev/null -g machines -c"descrizione_elaboratore_client" ↩
↪-s /bin/false nome_elaboratore$
```

```
# passwd -l nome_elaboratore$
```

```
# smbpasswd -a -m nome_elaboratore
```

È necessario prestare attenzione al carattere '**$**' alla fine del nome della macchina nel primo e nel secondo comando.
Il secondo comando permette di bloccare la parola d'ordine di quell'utente fittizio.
Con il terzo comando si definisce l'utente '***nome_elaboratore***' per Samba grazie all'opzione '**-m**'.

Con Samba 3 è possibile definire uno script per creare automaticamente gli utenti GNU/Linux e Samba corrispondenti alle utenze macchina; la direttiva da inserire in '`smb.conf`' è:

```
add machine script =
```

ad esempio:

```
add machine script = /usr/sbin/useradd -d /dev/null -g 100 -c descr -s /bin/false %u
```

6.7 Nota importante per clienti MS-Windows XP Professional

Nel caso si tenti di accedere da clienti MS-Windows XP (solo versione *Professional* in quanto con la versione *Home* non è possibile causa il mancato supporto dei domini), in presenza di serventi Samba non recenti, il processo di autenticazione può non concludersi positivamente a causa di una impostazione di sicurezza che deve essere variata nella macchina cliente.

A questo scopo occorre attivare la macchina MS-Windows XP come utente '**Administrator**' in locale ed effettuare i seguenti passaggi:

«Pannello di controllo»

«Prestazioni e manutenzione»

«Strumenti di amministrazione»

«Criteri di protezione locali»

«Criteri locali»

«Opzioni di protezione»

Nella lista che si ottiene si devono individuare le voci:

- «Membro di dominio: aggiunta crittografia o firma digitale ai dati del canale protetto (sempre)»
- «Membro di dominio: aggiunta crittografia o firma digitale ai dati del canale protetto (quando possibile)»
- «Membro di dominio: aggiunta firma digitale ai dati del canale protetto (quando possibile)»

e si devono impostare a «Disattivato».

Al successivo avvio della macchina sarà possibile farsi autenticare dal PDC Samba.

6.8 Nota importante per clienti MS-Windows 7

Nel caso si tenti di accedere da clienti MS-Windows 7, occorre fare in modo che tali macchine possano associarsi al dominio Samba intervenendo sul registro di sistema con il comando '**regedit**'.

Il percorso da considerare è '**HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\LanmanWorkstation\parameters**' e si devono aggiungere le seguenti due voci:

```
DWORD  DomainCompatibilityMode = 1
DWORD  DNSNameResolutionRequired = 0
```

Capitolo 7

Accesso a GNU/Linux da parte di utenti di una AD MS-Windows con Winbind

In precedenza, nel capitolo sull'impostazione di un servente Samba, è stato illustrato come integrare quest'ultimo in un dominio MS-Windows NT o in una *active directory* di MS-Windows.

In particolare si è visto come questo sia possibile impostando il livello di sicurezza al valore '**`domain`**' ed eseguendo il comando '**`net`**' o '**`smbpasswd`**' con opportune opzioni.

Grazie a Winbind, che è uno strumento di Samba presente dalla versione 2.2.2, diventa addirittura possibile l'autenticazione degli utenti GNU/Linux (attenzione: utenti GNU/Linux, non utenti Samba) presso un *controller di active directory* MS-Windows.

Ciò può essere molto utile in quei contesti in cui si vogliano inserire elaboratori con GNU/Linux in reti già consolidate su piattaforma MS-Windows, utilizzando le informazioni su utenti e gruppi preesistenti senza essere costretti a ridefinirle anche per le macchine GNU/Linux.

Winbind è costituito da un piccolo gruppo di componenti disponibili all'interno del pacchetto '**`samba-common`**'; in dettaglio ne fanno parte:

- una libreria per NSS (*Name Service Switch*);
- una libreria per i moduli PAM (*Pluggable Authentication Modules*);
- un programma di servizio, '**`wbinfo`**', e un demone, '**`winbindd`**'.

Il servizio NSS è presente in tutte le moderne librerie C e permette di ottenere i dati relativi a utenti, gruppi e nodi, da varie fonti (ad esempio NIS, DNS, ecc.); Winbind diventa un'ulteriore fonte di informazioni per NSS relativamente a utenti e gruppi di una *active directory* MS-Windows.

Il PAM è un sistema generalizzato per la gestione dei metodi di autenticazione per molteplici servizi (quelli per cui esistono le librerie PAM relative); grazie all'apposita libreria PAM, Winbind fornisce anche il servizio di autenticazione.

Per poter interagire con una *active directory* occorre poi che sulla macchina GNU/Linux siano installati i pacchetti relativi alla parte cliente e relativa configurazione di '**`Kerberos5`**' e cioè:

- '`krb5-user-`***`x.y.z`***`.deb`'
- '`krb5-conf-`***`x.y.z`***`.deb`'

In queste dispense non vengono forniti dettagli sul funzionamento e sulla configurazione di Kerberos ma solo le informazioni indispensabili affinché la macchina GNU/Linux possa ottenere il *ticket* di autenticazione dal servente Kerberos, cioè dal *controller* di *active directory*.

7.1 Configurazioni necessarie

Si suppone che il dominio di *active directory* al quale appartengono gli utenti sia ***planck.local*** e che il relativo servente MS-Windows si chiami ***win2003***.

7.1.1 /etc/krb5.conf

Iniziamo con la configurazione di '`/etckrb5.conf`':

```
[libdefaults]
  ticket_lifetime = 24000
  default_realm = PLANCK.LOCAL
  default_tkt_enctypes = des3-hmac-sha1 des-cbc-crc
  default_tgs_enctypes = des3-hmac-sha1 des-cbc-crc
[realms]
  PLANCK.LOCAL = {
   kdc = WIN2003
   admin_server = WIN2003
   default_domain = PLANCK.LOCAL
  }
[domain_realm]
  .domain.internal = PLANCK.LOCAL
  domain.internal = PLANCK.LOCAL
```

7.1.2 /etc/smb.conf

Occorre poi intervenire nel file '`/etc/smb.conf`' inserendo le direttive seguenti:

```
   workgroup = PLANCK
   realm = PLANCK.LOCAL
   encrypt password = yes
   security = ADS
   password server = win2003
   domain logons = no
; impostazioni per il demone winbindd
   winbind use default domain = yes
   winbind separator = +
   template shell = /bin/bash
   template homedir = /home/%D/%U
   winbind uid = 10000-20000
   winbind gid = 10000-20000
   winbind enum users = yes
   winbind enum groups = yes
```

Con '**`winbind separator`**' si imposta il carattere usato per ottenere il nome utente GNU/Linux dall'unione di nome di dominio e nome utente MS-Windows; il valore predefinito corrisponde a '\', ma è sconsigliabile, in quanto ha un significato speciale nella shell di GNU/Linux; la scelta del carattere '**+**' non dovrebbe invece comportare problemi.

Con '**`template shell`**' si imposta la shell degli utenti.

Con '**`template homedir`**' si definisce la directory personale degli utenti; nell'esempio si usano le variabili '**`%D`**' e '**`%U`**' in modo che ogni utente abbia come directory '`/home/`***nome_dominio_win***`/`***nome_utente_win***'.

'**`winbind uid`**' e '**`winbind gid`**' permettono di impostare gli intervalli di numeri di identificazione per utenti e gruppi che Winbind utilizza per riabbinare gli utenti e i gruppi MS-Windows a utenti e gruppi GNU/Linux.

'**winbind enum users**' e '**winbind enum groups**' permettono di attivare l'enumerazione di gruppi e utenti.

7.1.3 Modifiche ai file di configurazione dei moduli PAM

Le modifiche ai file di configurazione dei moduli PAM devono essere effettuate con molta attenzione in quanto errori in questa fase possono anche causare l'impossibilità di accedere al sistema. Può quindi essere opportuno fare una copia dei file interessati alle modifiche in modo da poter ripristinare la situazione precedente in ogni momento e anche tenere aperta una sessione di riserva come utente '**root**' dalla quale intervenire in caso di necessità.

Maggiori informazioni sul funzionamento dei moduli PAM non possono essere fornite in questa sede.

In testa al file '`/etc/pam.d/common-account`' deve essere aggiunta la seguente riga:

```
account  sufficient /lib/security/pam_winbind.so
```

In testa al file '`/etc/pam.d/common-auth`' deve essere aggiunta la seguente riga:

```
account  sufficient /lib/security/pam_winbind.so
```

Inoltre cambiare la riga:

```
auth   [success=1 default=ignore]   pam_unix.so nullok_secure
```

in:

```
auth   [success=1 default=ignore]   pam_unix.so nullok_secure use_first_pass
```

In testa al file '`/etc/pam.d/common-password`' deve essere aggiunta la seguente riga:

```
account  sufficient /lib/security/pam_winbind.so
```

In testa al file '`/etc/pam.d/common-session`' deve essere aggiunta la seguente riga:

```
session  required   /lib/security/pam_mkhomedir.so  skel=/etc/skel/  umask=0002
```

Questa è molto importante in quanto permette la creazione automatica della directory personale dell'utente al primo accesso alla macchina GNU/Linux secondo quanto inserito nel file '`smb.conf`':

```
template homedir = /home/%D/%U
```

7.1.4 Modifiche alla configurazione di NSS

Nel file '`/etc/nsswitch.conf`', contenente la configurazione del servizio NSS, è necessario aggiungere Winbind tra le fonti dei dati relativi a utenti e gruppi. Ad esempio:

```
passwd:     files winbind
group:       files winbind
```

L'ordine con cui vengono elencate le fonti è significativo e quindi è opportuno lasciare la priorità a '**files**' oppure a '**compat**' in modo che per primi siano interrogati i file di sistema ('`/etc/passwd`' e '`/etc/group`').

7.2 Attivazione

Per prima cosa occorre ottenere il *ticket* dal servente MS-Windows con il comando:

```
# kinit Administrator@PLANCK.LOCAL
```

cui segue la richiesta della parola d'ordine di ***Administrator*** (che si suppone essere un utente amministrativo del *domain controller*).

Affinché l'autenticazione Kerberos funzioni correttamente occorre che tutte le macchine coinvolte siano sincronizzate relativamente all'orario di sistema.

Si può verificare il buon esito del comando precedente con:

```
# klist
```

che fornisce la lista dei *ticket* presenti nella relativa *cache*.

A questo punto si deve inserire la macchina GNU/Linux nella *active directory* MS-Windows:

- sul servente MS-Windows 200x con la MMC (*Microsoft Management Console*) attraverso lo strumento «Utenti e computer di Active Directory», aggiungere il servente Samba alla *active directory*.
- eseguire quindi, sulla macchina GNU/Linux, il comando seguente:

  ```
  # net ads join -UAdministrator%parola_d'ordine -Swin2003.planck.local
  ```

Affinché il comando abbia successo occorre anche che il nome del *controller* della *active directory* sia risolvibile; in caso non si abbia a disposizione un servente DNS basta aggiungere al file '`/etc/hosts`' una riga come la seguente (dove si suppone che ***192.168.56.101*** sia il numero IP della macchina *controller*):

```
192.168.56.101   win2003.planck.local   win2003
```

Infine si devono avviare i servizi '**samba**' e '**winbind**':

```
# /etc/rc.d/init.d/samba start

# /etc/rc.d/init.d/winbind start
```

Per verificare il buon funzionamento di Winbind eseguire i comandi:

```
# wbinfo -u

# wbinfo -g
```

con i quali si elencano rispettivamente utenti e gruppi della *active directory* MS-Windows (vedere la figura 7.11).

Figura 7.11.

Si possono usare anche i comandi:

```
# getent passwd
```

```
# getent group
```

per ottenere gli elenchi di tutti gli utenti e gruppi utilizzabili, sia quelli della *active directory* che quelli propri di GNU/Linux.

Infine si può procedere all'accreditamento sulla macchina GNU/Linux di un utente della *active directory* MS-Windows (vedere figure 7.12 e 7.13 notando in 7.12 la creazione automatica della directory dell'utente al primo accesso).

Figura 7.12.

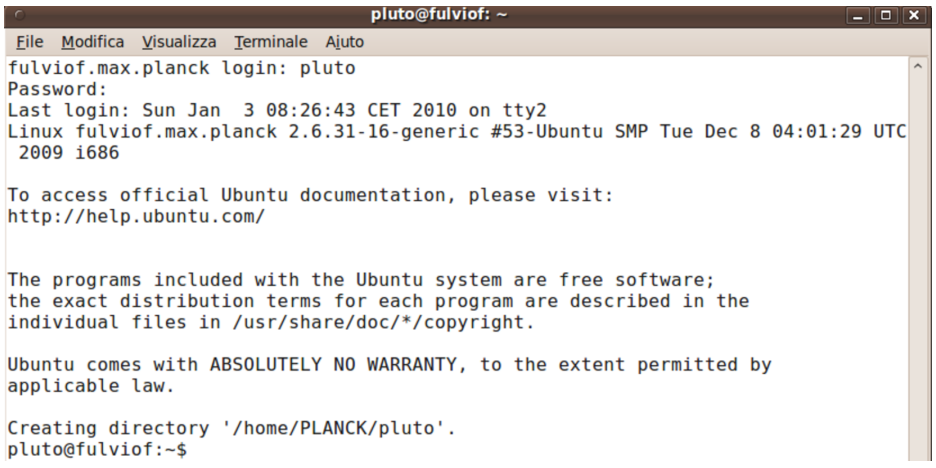


Figura 7.13.

7.3 Esportazione delle directory utente dal servente MS-Windows

Può essere interessante fare in modo che le directory personali dei vari utenti che si accreditano sui clienti GNU/Linux siano fisicamente residenti sulla macchina MS-Windows e vengano esportate al momento dell'avvio delle macchine GNU/Linux.

Per ottenere ciò basta inserire nel file '`/etc/fstab`' di ogni macchina GNU/Linux la riga seguente:

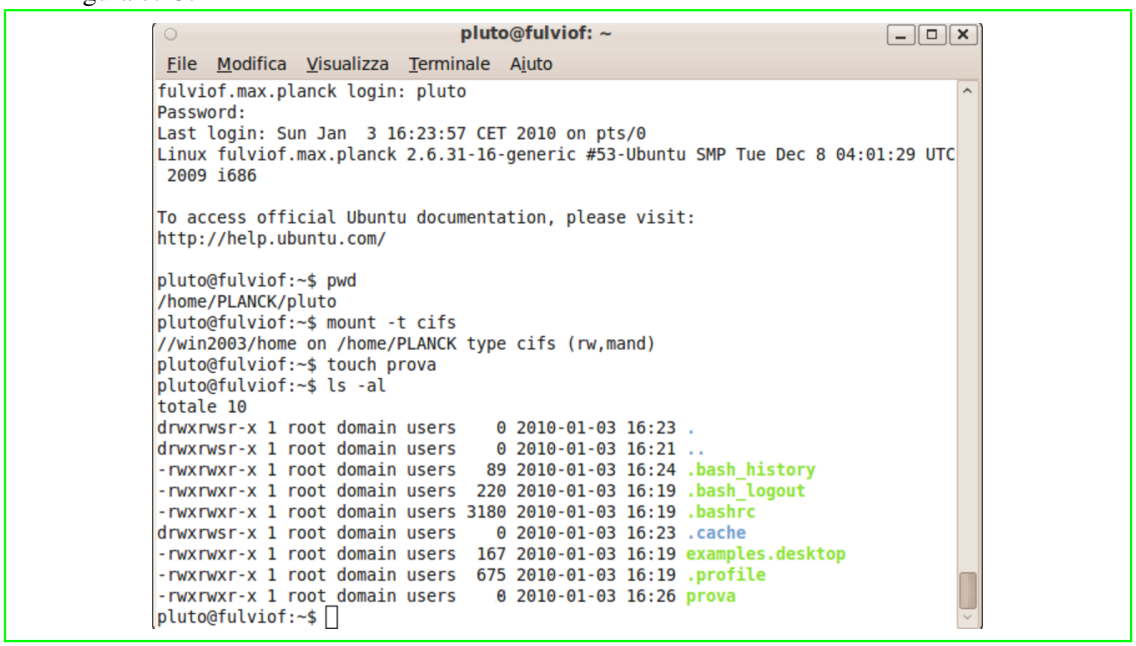
```
//win2003/home    /home/PLANCK/    cifs    username=Administrator%parola_d'ordine,↩
↪gid=Domain\ Users,dir_mode=02775,file_mode=0775
```

Si suppone che sui clienti le directory personali vengano create in '`/home/PLANCK/`', come mostrato in tutti gli esempi precedenti, e che esse risiedano sul servente come risorsa condivisa ***home*** (cui si devono ovviamente assegnare gli opportuni permessi in ambiente MS-Windows).

Grazie alle opzioni impostate in sede di montaggio ('**`gid, dir_mode, file_mode`**') le directory personali risultano pienamente gestibili dagli utenti membri del gruppo ***Domain Users*** e quindi dai rispettivi proprietari (vedere figura. 7.15).

Si presti però attenzione al fatto che qualsiasi utente della *active directory*, una volta accreditatosi in GNU/Linux, ha il controllo su tutte le directory personali disponibili in '`/home/PLANCK`', cosa che rende questa soluzione praticabile solo a livello didattico e comunque in contesti in cui non siano importanti la sicurezza e la *privacy* dei dati.

Figura 7.15.



Samba e DFS

Il DFS (*Distributed File System*), introdotto con MS-Windows 2000, permette di organizzare le condivisioni di rete in una struttura ad albero svincolando gli utenti di tali risorse dalla conoscenza della reale collocazione delle stesse sui vari serventi. Con MS-Windows 95/98/NT è invece necessario, quando si deve connettere una risorsa, conoscere esattamente la sua collocazione in rete; si può ovviare in parte a questo inconveniente connettendo permanentemente la risorsa all'avvio dell'elaboratore cliente, ma in caso di spostamento della risorsa il problema si ripresenta.

Con il DFS si vengono a creare dei «volumi di rete» che possono essere ispezionati come fossero residenti fisicamente su un solo servente. La struttura può poi essere duplicata, tutta o in parte, per ottenere maggiori garanzie contro le perdite di dati accidentali.

Ogni struttura DFS ha una radice comune a tutte le condivisioni e numerose diramazioni (foglie), tutte di primo livello. Un servente può ospitare una sola radice mentre le foglie possono anche essere ospitate su macchine diverse. Sarebbe anche possibile ottenere strutture più complesse annidando radici di DFS come foglie di altri DFS ma qui non si considera tale possibilità.

Samba può assumere il ruolo di servente DFS e ospitare una radice di un volume DFS grazie alla seguente direttiva nella sezione '**`global`**':

```
    host msdfs = yes
```

e alla definizione di questa nuova sezione:

```
[dfs]
   path = /dir-expo/dfs
   msdfs root = yes
```

Nella directory '`/dir-expo/dfs`' del servente GNU/Linux si dovranno poi impostare i collegamenti simbolici agli altri serventi della rete procedendo come nell'esempio seguente:

```
# cd /dir-expo/dfs

# chown root /dir-expo/dfs

# chmod 755 /dir-expo/dfs

# ln -s msdfs:servA\\shareA coll-a

# ln -s msdfs:servB\\shareBC,servC\\shareBC coll-bc
```

Il secondo collegamento dell'esempio associa a un solo nome di risorsa DFS, due condivisioni: queste saranno in *fault tolerance* tra di loro e l'allineamento dei dati al loro interno sarà assicurato dal servente DFS.

Grazie alla definizione dei collegamenti simbolici, quando un cliente si collega a una risorsa DFS, viene ridiretto, in modo del tutto trasparente e automatico, verso la macchina che ospita fisicamente i dati condivisi.

Capitolo 9

Samba e OpenLDAP: gestione centralizzata degli utenti di rete

In questo capitolo viene esaminata la possibilità di utilizzare Samba come PDC con gli utenti della rete memorizzati in un archivio gestito da OpenLDAP [1] (⟨*http://www.openldap.org*⟩).

Uno dei principali vantaggi di una soluzione di questo genere è che anche eventuali macchine clienti GNU/Linux possono far accreditare gli utenti servendosi dello stesso archivio; è così possibile una gestione completamente centralizzata e uniforme degli utenti in una rete eterogenea.

Prima di entrare nei dettagli vengono presi in esame la natura del servizio LDAP e la sua installazione e configurazione indipendentemente da Samba.

9.1 Il servizio LDAP

LDAP (*Lightweight Directory Access Protocol*) è un protocollo di tipo cliente/servente proposto nel 1995 dall'università del Michigan allo scopo di accedere alle «directory» o «elenchi» in rete.

Da qui in poi si fa uso del termine «elenchi», anche se è sicuramente meno usato, soprattutto per evitare possibili confusioni con l'altro significato che il termine «directory» ha relativamente alla organizzazione logica dei dati sulle memorie di massa.

Nella descrizione dell'acronimo LDAP si nota che è presente il termine «lightweight» che fa pensare a qualcosa di poco pesante; il motivo è che questo protocollo si pone come un'alternativa semplificata al preesistente protocollo ISO/OSI (*International Organization for Standardization/Open Systems Interconnection*) X.500 definito per la gestione degli elenchi in rete e noto anche come protocollo DAP.

I vantaggi di LDAP rispetto a X.500 sono fondamentalmente i seguenti:

- X.500 non è compatibile con Internet, LDAP invece si perché usa TCP/IP;
- X.500 è molto più complesso rispetto a LDAP;
- LDAP è comunque compatibile con gli elenchi definiti secondo le specifiche X.500.

Esistono vari servizi di elenchi in rete proprietari, come ad esempio Oracle Internet Directory o Microsoft Active Directory; con OpenLDAP, del quale ci occupiamo in questa sede, è possibile realizzarli in modo completamente libero.

9.1.1 Natura degli elenchi in rete

Un elenco è un catalogo di informazioni anche eterogenee organizzate in una struttura gerarchica; secondo la terminologia ufficiale si parla di un DIT (*Directory Information Tree*) che contiene vari nodi (*objectclass*) ognuno dei quali ha una serie di attributi obbligatori o facoltativi che forniscono informazioni sul nodo.

La struttura gerarchica di un elenco può essere rappresentata come un albero capovolto, con la radice in alto; in un elenco possono comunque essere presenti più alberi (cataloghi diversi).

Ogni nodo di un albero può avere un solo genitore e un numero variabile di figli.

Per ogni elenco deve essere definito uno '**`schema`**' cioè l'insieme degli attributi possibili dei nodi, della loro sintassi e dei tipi di oggetti presenti.

In base a questa prima descrizione si può pensare di confrontare un elenco con una base di dati in quanto in entrambi i casi si tratta di insiemi strutturati di dati; esistono però delle differenze che sono di fondamentale importanza:

- i servizi di elenchi sono ottimizzati per il solo reperimento di informazioni e quindi adatti solamente nel caso di archivi molto statici, questo ovviamente non è vero per le basi di dati;
- negli elenchi è generalmente possibile immagazzinare informazioni in modo più descrittivo rispetto alle basi di dati;
- negli elenchi le informazioni possono essere agevolmente separate in alberi distinti, anche residenti su serventi diversi, così da avere, in modo semplice, distribuzione dei dati;
- in modo altrettanto semplice è possibile replicare i dati di un elenco su più serventi così da avere ridondanza;
- negli elenchi sono ammesse temporanee inconsistenze sui dati (dovute ad esempio ad aggiornamenti non istantanei tra serventi diversi), cosa non possibile in qualsiasi base di dati;
- per gli elenchi non esiste o non è importante il concetto di transazione (o di operazione «atomica»), visto anche che gli accessi ai dati sono quasi esclusivamente in lettura;
- per l'accesso ai dati di un elenco non è necessario usare un linguaggio strutturato come SQL (*Structured Query Language*) ma è sufficiente rispettare le regole semplificate definite dal protocollo LDAP.

9.1.2 LDAP per gestire gli utenti di una rete

Uno degli utilizzi più interessanti di LDAP è la gestione centralizzata degli utenti di una rete in ambiente GNU/Linux, compito per il quale può validamente sostituire il vecchio servizio NIS (*Network Information Service*).

Rispetto al NIS infatti LDAP ha una importante serie di vantaggi:

- LDAP può funzionare «a cavallo» di due reti, NIS no;
- LDAP, a differenza di NIS, è adatto anche in presenza di una mole considerevole di utenti;
- con NIS le informazioni circolano sempre in chiaro, con LDAP non necessariamente in quanto è possibile usare strumenti come TSL/SSL per avere trasmissione cifrata dei dati, autenticazione dei nodi di rete, verifica di integrità dei dati;
- le informazioni gestite in un elenco LDAP possono essere usate anche per altri scopi oltre che per l'autenticazione degli utenti (ad esempio per un indirizzario di posta elettronica), con NIS questo non è possibile;
- con LDAP si possono concentrare in un unico archivio: gli utenti della rete su clienti GNU/Linux e quelli di un PDC Samba e quindi anche dei clienti MS-Windows; con NIS questo non è possibile: si hanno utenti GNU/Linux e utenti MS-Windows su due archivi separati con necessità di mantenere l'allineamento (cosa non banale da realizzare).

Quanto appena affermato è realizzabile se Samba è configurato per utilizzare LDAP come sistema di memorizzazione delle parole d'ordine (cioè come *password backend*). Questo è proprio l'argomento per il quale è stato scritto il presente capitolo.

9.1.3 Definizione di un elenco

Prima di illustrare l'installazione e la configurazione di OpenLDAP è opportuno soffermarsi sulla progettazione dell'elenco delle informazioni da gestire.

Gli oggetti presenti in un elenco che sono «foglie» di un albero, cioè nodi terminali, senza figli, sono raggiungibili grazie ad un identificatore univoco chiamato DN (*Distingueshed Name*) che ha un aspetto simile al seguente:

```
uid=rossimario,ou=People,dc=max,dc=planck
```

Leggendo il DN da sinistra a destra si ottiene il percorso per risalire dalla foglia alla radice dell'albero; in senso inverso si ottiene la scala gerarchica dei nodi presenti.

Nell'esempio abbiamo due DC (*Domain Component*) con valori rispettivamente ***max*** e ***planck*** che rappresentano il suffisso (o radice) dell'albero delle informazioni; seguono poi OU (*Organizational Unit*) con valore ***People*** e UID (User id) con valore ***rossimario***.

In pratica vuol dire che nell'elenco esiste un nodo identificato dall'uid ***rossimario***, appartenente al gruppo o settore delle persone, all'interno dell'organizzazione identificata dal suffisso ***max.planck***.

Naturalmente non è detto che tutti i DN siano definiti con le stesse componenti in quanto dipendono dal tipo di nodo cui si riferiscono; sicuramente avranno questa struttura i DN relativi a oggetti di tipo ***People***.

I valori degli attributi degli oggetti di un elenco non sono *case sensitive*.

Esiste anche un altro tipo di identificatore chiamato RDN (*Relative Distingueshed Name*) che può essere associato a qualsiasi nodo ma che non costituisce il percorso completo per reperirlo; esempi possono essere:

```
uid=rossimario
ou=People
```

La decisione su quale sia la radice dell'albero delle informazioni è ovviamente molto importante e deve essere presa con attenzione; esistono due metodi che solitamente vengono usati nella definizione degli elenchi in rete:

1. suffisso che identifica una organizzazione (azienda, scuola o altro) definito partendo da un nome di dominio; nell'esempio è stato scelto questo metodo usando un dominio privato, cioè non presente in Internet;
2. suffisso definito con un criterio geografico come nell'esempio che segue.

```
ou=finance,o=Government,st=Minnesota,c=US
```

In questo caso abbiamo un nodo che potrebbe essere relativo al settore finanze della organizzazione ***Govern*** dello stato ***Minnesota*** negli Stati Uniti.

Il primo metodo è comunque solitamente preferito e viene utilizzato anche nel proseguo di queste dispense mantenendo il nome ***max.planck***.

Come detto in precedenza ogni oggetto dell'elenco ha una serie di attributi grazie ai quali si rendono disponibili le informazioni relative a quell'oggetto.

Facendo di nuovo riferimento all'esempio di un oggetto di tipo ***People*** potremmo avere degli attributi come:

```
cn=Rossi Mario
mail=rossimario@max.planck
```

dove CN significa *Common Name*.

Per sapere quali sono i tipi di oggetti presenti in un elenco e quali sono i relativi attributi si deve conoscere quale è lo schema dell'elenco.

Per fortuna non è quasi mai necessario definire gli schemi degli elenchi da parte dell'amministratore degli stessi, in quanto nel pacchetto OpenLDAP sono già inseriti degli schemi precompilati che sono sufficienti per tutti i tipi più comuni di elenchi in rete.

9.2 Usare OpenLDAP

In questo paragrafo vediamo come installare, configurare e usare OpenLDAP versione 2.4 per la gestione degli utenti, facendo riferimento alla distribuzione Ubuntu.

Non vengono prese in considerazione alcune funzionalità come la replica dell'archivio su serventi diversi.

9.2.1 Installazione di OpenLDAP

Per l'installazione dei pacchetti si può utilizzare uno degli strumenti tipici dell'ambiente Debian come '**`apt-get`**' da linea di comando o '**`synaptic`**' dalla grafica; qui non vengono forniti dettagli sull'utilizzo di questi strumenti per i quali si rimanda alle numerose fonti disponibili in Internet.

I pacchetti da installare sono:

- '`slapd_`***x.y.z***`.deb`'
- '`libldap2_`***x.y.z***`.deb`'
- '`ldap-utils_`***x.y.z***`.deb`'

Il pacchetto '**`slapd`**' contiene il servente OpenLDAP e deve essere installato soltanto nella macchina servente.

Gli altri due pacchetti, che contengono rispettivamente le librerie per l'accesso agli elenchi e alcuni programmi di utilità per la gestione degli stessi, devono essere invece installati sia nel servente che nei clienti.

Nel pacchetto '**`slapd`**' è presente anche il demone '**`slurpd`**' che serve a gestire le repliche dell'archivio OpenLDAP tra serventi diversi e che qui, come detto, non usiamo.

Altri pacchetti da usare per sfruttare i servizi offerti da OpenLDAP; vengono qui solo elencati in attesa di esaminare i dettagli al momento opportuno:

- '`migrationtools_`***x.y.z***`.deb`'

  contiene dei programmi scritti in Perl per la migrazione delle informazioni (utenti, gruppi ecc.) dagli archivi GNU/Linux (file '`passwd`', '`group`' ecc.) all'elenco OpenLDAP;

- ‘`smbldap-tools_`***x.y.z***`.deb`’

  contiene i programmi scritti in Perl per la gestione degli utenti Samba in OpenLDAP;

- ‘`ldap-auth-client_`***x.y.z***`.deb`’
- ‘`ldap-auth-config_`***x.y.z***`.deb`’
- ‘`auth-client-config_`***x.y.z***`.deb`’
- ‘`libnss-ldap_`***x.y.z***`.deb`’
- ‘`libpam-ldap_`***x.y.z***`.deb`’

  usati per permettere l’accreditamento nel sistema GNU/Linux agli utenti gestiti con OpenLDAP;

- ‘`samba-doc_`***x.y.z***`.deb`’

  contiene lo schema LDAP per Samba;

- ‘`nscd_`***x.y.z***`.deb`’

  per il *caching* delle informazioni riguardanti l’accreditamento degli utenti (permette di migliorare le prestazioni di servizi «lenti» come NIS e OpenLDAP.

Siccome alcuni dei pacchetti citati comprendono programmi scritti in Perl, è necessario che quest’ultimo sia installato nella macchina che stiamo utilizzando.

9.2.2 Configurazione di OpenLDAP

Nelle ultime versioni di OpenLDAP (precisamente a partire dalla ‘**`2.3`**’) il sistema di configurazione è completamente variato.

In precedenza veniva usato il classico file di configurazione ‘`/etc/ldap/slapd.conf`’ (che comunque può ancora essere utilizzato, come descritto nel paragrafo 9.2.3), mentre ora si è passati ad un metodo basato su una voce DIT speciale che contiene i vari parametri e che viene denominato in varie maniere:

- ‘**`cn=config configuration`**’; dal nome della voce dell’albero di directory che contiene i parametri;
- ‘**`slapd.d configuration`**’; dal nome della directory che contiene fisicamente l’archivio con i dati di configurazione all’interno di ‘`/etc/ldap`’;
- ‘**`run-time configuration`**’ o ‘**`zero down-time configuration`**’; dal fatto che non occorre più fermare e riavviare il servizio dopo ogni variazione nella configurazione.

Il nuovo sistema è stato adottato proprio per avere maggiore flessibilità e per poter variare dinamicamente la configurazione con il servizio attivo.

Con alcune distribuzioni di GNU/Linux c’è la possibilità di inserire i parametri fondamentali per l’uso di OpenLDAP semplicemente riconfigurando il pacchetto con il comando:

```
# dpkg-reconfigure slapd
```

In questo modo si ricevono varie schermate con le richieste relative (tra le altre) alla radice dell’elenco e ai dati di accreditamento dell’utente amministratore.

Nelle versioni più recenti di Ubuntu questo non avviene in quanto il servente viene installato con una configurazione minima e deve essere manipolato con i comandi messi a disposizione dal pacchetto '**ldap-utils**'.

Gli unici aspetti su cui si può intervenire con la riconfigurazione sono:

- la cancellazione dell'archivio in caso di eliminazione del pacchetto (si consiglia la risposta affermativa);
- l'attivazione del supporto al vecchio protocollo LDAPv2 (la versione attuale è la 3); si può rispondere negativamente a meno che non si abbiano programmi che lo utilizzano.

Di seguito vediamo come inserire le informazioni di base nell'archivo di OpenLDAP, in una versione recente di Ubuntu.

Allo scopo vengono usati il comando '**ldapadd**' e i file con formato '**LDIF**', che vengono dettagliati maggiormente nel paragrafo 9.2.5:

1. rendere disponibili gli schemi fondamentali (presenti, dopo l'installazione, nella posizione '`/etc/ldap/schema/`'), con il comando:

   ```
   # ls /etc/ldap/schema/*.ldif | xargs -I {} ldapadd -Y EXTERNAL↩
   ↪-H ldapi:/// -f {}
   ```

2. definire il suffisso dell'elenco (che si suppone corrisponda al dominio '**max.planck**'), il tipo di archivio (BDB *Berkeley Data Base*) e i dati dell'amministratore, con il comando:

   ```
   # ldapadd -Y EXTERNAL -H ldapi:/// -f slapd_db.ldif
   ```

   dopo avere creato il file '`slapd_db.ldif`' con il seguente contenuto:

```
# Carica i moduli per il tipo di db
dn: cn=module,cn=config
objectclass: olcModuleList
cn: module
olcModuleLoad: back_bdb.la
# Crea l'archivio directory
dn: olcDatabase=bdb,cn=config
objectClass: olcDatabaseConfig
objectClass: olcBdbConfig
olcDatabase: bdb
# Nome di dominio (esempio max.planck)
olcSuffix: dc=max,dc=planck
# Percorso che ospita il db
olcDbDirectory: /var/lib/ldap
# Dati dell'amministratore
olcRootDN: cn=admin,dc=max,dc=planck
# Ottenere la parola d'ordine con il comando: "slappasswd -h {crypt}" che
# chiede due volte la password in chiaro (supponiamo sia admin)
# e poi incollarla nella riga seguente
olcRootPW: {CRYPT}g4GGeu14vX3xA
# Indici per velocizzare le ricerche
olcDbIndex: uid pres,eq
olcDbIndex: cn,sn,mail pres,eq,approx,sub
olcDbIndex: objectClass eq
# Permette all'utente di modificare la propria psw
# Permette all'utente anonimo di autenticarsi
# Permette ad admin di cambiare la psw di chiunque
olcAccess: to attrs=userPassword
  by self write
  by anonymous auth
  by dn.base="cn=admin,dc=max,dc=planck" write
  by * none
# Permette agli utenti di cambiare i propri dati
# Permette a tutti di leggere la directory
olcAccess: to *
  by self write
  by dn.base="cn=admin,dc=max,dc=planck" write
  by * read
```

3. inserire la radice dell'elenco qualche unità organizzativa di base, con il comando:

   ```
   # ldapadd -x -D cn=admin,dc=max,dc=planck -w admin↩
   ↪-f slapd_people.ldif
   ```

   dove si suppone che la *password* dell'amministratore sia ***admin***.

   Il file '`slapd_people.ldif`' deve essere creato con il seguente contenuto:

```
# Create top-level object in domain
dn: dc=max,dc=planck
objectClass: top
objectClass: dcObject
objectclass: organization
o: max.planck
dc: max
description: Home network
dn: ou=people,dc=max,dc=planck
objectClass: organizationalUnit
ou: people
dn: ou=groups,dc=max,dc=planck
objectClass: organizationalUnit
ou: groups
```

Per verificare l'esito positivo delle operazioni appena descritte si può eseguire il comando:

```
# slapcat
```

che serve ad estrarre le informazioni dall'elenco e dovrebbe fornire questo risultato:

```
dn: dc=max,dc=planck
objectClass: top
objectClass: dcObject
objectClass: organization
o: max.planck
dc: max
description:: SG9tZSBuZXR3b3JrIA==
structuralObjectClass: organization
entryUUID: b59b4738-f605-102e-9bfe-8f074a5e4515
creatorsName: cn=admin,dc=max,dc=planck
createTimestamp: 20100517134158Z
entryCSN: 20100517134158.581185Z#000000#000#000000
modifiersName: cn=admin,dc=max,dc=planck
modifyTimestamp: 20100517134158Z

dn: ou=people,dc=max,dc=planck
objectClass: organizationalUnit
ou: people
structuralObjectClass: organizationalUnit
entryUUID: b59be2f6-f605-102e-9bff-8f074a5e4515
creatorsName: cn=admin,dc=max,dc=planck
createTimestamp: 20100517134158Z
entryCSN: 20100517134158.585171Z#000000#000#000000
modifiersName: cn=admin,dc=max,dc=planck
modifyTimestamp: 20100517134158Z

dn: ou=groups,dc=max,dc=planck
objectClass: organizationalUnit
ou: groups
structuralObjectClass: organizationalUnit
entryUUID: b59c1a82-f605-102e-9c00-8f074a5e4515
creatorsName: cn=admin,dc=max,dc=planck
createTimestamp: 20100517134158Z
entryCSN: 20100517134158.586593Z#000000#000#000000
modifiersName: cn=admin,dc=max,dc=planck
modifyTimestamp: 20100517134158Z
```

Concludiamo questo paragrafo sulla configurazione mostrando il contenuto di un altro file, molto importante, presente in Ubuntu e altre distribuzioni derivate da Debian; si tratta del file '`/etc/default/slapd`' che contiene alcune impostazioni fondamentali come il nome dell'utente e del gruppo con i privilegi dei quali è in esecuzione il servente OpenLDAP e soprattutto l'indicazione di quale è la modalità di configurazione adottata.

Quest'ultima impostazione è visibile proprio all'inizio del file; non aggiungiamo ulteriori considerazioni sul suo contenuto in quanto i commenti già presenti sono sufficientemente chiari.

```
1  # Location of the slapd configuration to use.  If using the cn=config
2  # backend to store configuration in LDIF, set this variable to the
3  # directory containing the cn=config data; otherwise set it to the location
4  # of your slapd.conf file.  If empty, use the compiled-in default
5  # (/etc/ldap/slapd.d).
6  SLAPD_CONF=
7
8  # System account to run the slapd server under. If empty the server
```

```
9   # will run as root.
10  SLAPD_USER="openldap"
11
12  # System group to run the slapd server under. If empty the server will
13  # run in the primary group of its user.
14  SLAPD_GROUP="openldap"
15
16  # Path to the pid file of the slapd server. If not set the init.d script
17  # will try to figure it out from $SLAPD_CONF (/etc/ldap/slapd.d by
18  # default)
19  SLAPD_PIDFILE=
20
21  # slapd normally serves ldap only on all TCP-ports 389. slapd can also
22  # service requests on TCP-port 636 (ldaps) and requests via unix
23  # sockets.
24  # Example usage:
25  # SLAPD_SERVICES="ldap://127.0.0.1:389/ ldaps:/// ldapi:///"
26  SLAPD_SERVICES="ldap:/// ldapi:///"
27
28  # If SLAPD_NO_START is set, the init script will not start or restart
29  # slapd (but stop will still work).  Uncomment this if you are
30  # starting slapd via some other means or if you don't want slapd normally
31  # started at boot.
32  #SLAPD_NO_START=1
33
34  # If SLAPD_SENTINEL_FILE is set to path to a file and that file exists,
35  # the init script will not start or restart slapd (but stop will still
36  # work).  Use this for temporarily disabling startup of slapd (when doing
37  # maintenance, for example, or through a configuration management system)
38  # when you don't want to edit a configuration file.
39  SLAPD_SENTINEL_FILE=/etc/ldap/noslapd
40
41  # For Kerberos authentication (via SASL), slapd by default uses the system
42  # keytab file (/etc/krb5.keytab).  To use a different keytab file,
43  # uncomment this line and change the path.
44  #export KRB5_KTNAME=/etc/krb5.keytab
45
46  # Additional options to pass to slapd
47  SLAPD_OPTIONS=""
```

9.2.3 Vecchio sistema di configurazione di OpenLDAP

Come accennato in precedenza, il vecchio metodo di configurazione del servizio OpenLDAP è ancora utilizzabile; prima di vedere come, esaminiamo il contenutodi un file di configurazione '`slapd.conf`' relativo ad una versione di OpenLDAP precedente alla '**`2.3`**':

```
1  # This is the main slapd configuration file. See slapd.conf(5) for more
2  # info on the configuration options.
3
4  ########################################################################
5  # Global Directives:
6
```

```
7   # Features to permit
8   #allow bind_v2
9
10  # Schema and objectClass definitions
11  include         /etc/ldap/schema/core.schema
12  include         /etc/ldap/schema/cosine.schema
13  include         /etc/ldap/schema/nis.schema
14  include         /etc/ldap/schema/inetorgperson.schema
15
16  # Schema check allows for forcing entries to
17  # match schemas for their objectClasses's
18  # schemacheck     on
19
20  # Where the pid file is put. The init.d script
21  # will not stop the server if you change this.
22  pidfile         /var/run/slapd/slapd.pid
23
24  # List of arguments that were passed to the server
25  argsfile        /var/run/slapd/slapd.args
26
27  # Read slapd.conf(5) for possible values
28  loglevel        256
29
30  # Where the dynamically loaded modules are stored
31  modulepath   /usr/lib/ldap
32  moduleload   back_bdb
33
34  # The maximum number of entries that is returned for a search operation
35  sizelimit 500
36
37  # The tool-threads parameter sets the actual amount of cpu's that is used
38  # for indexing.
39  tool-threads 1
40
41  ########################################################################
42  # Specific Backend Directives for bdb:
43  # Backend specific directives apply to this backend until another
44  # 'backend' directive occurs
45  backend          bdb
46  # checkpoint 512 30
47
48  ########################################################################
49  # Specific Backend Directives for 'other':
50  # Backend specific directives apply to this backend until another
51  # 'backend' directive occurs
52  #backend         <other>
53
54  ########################################################################
55  # Specific Directives for database #1, of type bdb:
56  # Database specific directives apply to this databasse until another
57  # 'database' directive occurs
58  database        bdb
59
60  # The base of your directory in database #1
```

```
61  suffix          "dc=max,dc=planck"
62
63  # Where the database file are physically stored for database #1
64  directory       "/var/lib/ldap"
65
66  # For the Debian package we use 2MB as default but be sure to update this
67  # value if you have plenty of RAM
68  dbconfig set_cachesize 0 2097152 0
69
70  # Sven Hartge reported that he had to set this value incredibly high
71  # to get slapd running at all. See http://bugs.debian.org/303057
72  # for more information.
73
74  # Number of objects that can be locked at the same time.
75  dbconfig set_lk_max_objects 1500
76  # Number of locks (both requested and granted)
77  dbconfig set_lk_max_locks 1500
78  # Number of lockers
79  dbconfig set_lk_max_lockers 1500
80
81  # Indexing options for database #1
82  index           objectClass eq
83
84  # Save the time that the entry gets modified, for database #1
85  lastmod         on
86
87  # Where to store the replica logs for database #1
88  # replogfile   /var/lib/ldap/replog
89
90  # The userPassword by default can be changed
91  # by the entry owning it if they are authenticated.
92  # Others should not be able to see it, except the
93  # admin entry below
94  # These access lines apply to database #1 only
95  access to attrs=userPassword,shadowLastChange
96          by dn="cn=admin,dc=max,dc=planck" write
97          by anonymous auth
98          by self write
99          by * none
100
101 # Ensure read access to the base for things like
102 # supportedSASLMechanisms.  Without this you may
103 # have problems with SASL not knowing what
104 # mechanisms are available and the like.
105 # Note that this is covered by the 'access to *'
106 # ACL below too but if you change that as people
107 # are wont to do you'll still need this if you
108 # want SASL (and possible other things) to work
109 # happily.
110 access to dn.base="" by * read
111
112 # The admin dn has full write access, everyone else
113 # can read everything.
114 access to *
```

```
115         by dn="cn=admin,dc=max,dc=planck" write
116         by * read
117
118 # For Netscape Roaming support, each user gets a roaming
119 # profile for which they have write access to
120 #access to dn=".*,ou=Roaming,o=morsnet"
121 #        by dn="cn=admin,dc=max,dc=planck" write
122 #        by dnattr=owner write
123
124 ########################################################################
125 # Specific Directives for database #2, of type 'other' (can be bdb too):
126 # Database specific directives apply to this databasse until another
127 # 'database' directive occurs
128 #database        <other>
129
130 # The base of your directory for database #2
131 #suffix   "dc=debian,dc=org"
```

La numerazione delle righe ovviamente non è presente nel file, è stata aggiunta per poter fare riferimento, in modo più preciso, alle righe di maggiore interesse.

Il file contiene righe di commento che, come al solito, iniziano con «#» ed è suddiviso in tre sezioni:

- direttive globali;
- specifiche del *backend*
- specifiche della base di dati.

Le opzioni presenti in una sezione possono sovrascrivere quelle presenti in una sezione precedente.

Nella sezione delle direttive globali vediamo la riga 8 che, se non commentata, permette di avere il supporto per il vecchio protocollo LDAPv2; a seguire una serie di righe (dalla 11 alla 14) che permettono di includere nella definizione del nostro elenco alcuni schemi già pronti.

Questi schemi sono di solito sufficienti per un uso «normale» di OpenLDAP (eventualmente sarebbe da aggiungere lo schema per Samba in caso si vogliano avere gli utenti di quest'ultimo gestiti in un elenco).

Nel caso fosse necessario utilizzare ulteriori schemi, anche di nostra produzione, basta aggiungere la relativa riga '**`include`**' nel file di configurazione e il file contenente le specifiche dello schema nella directory '`/etc/apt/schema`'.

A titolo di esempio viene mostrata una porzione del file '`/etc/ldap/schema/core.schema`' riguardante gli attributi ***uid*** e ***mail***:

```
attributetype ( 0.9.2342.19200300.100.1.1
        NAME ( 'uid' 'userid' )
        DESC 'RFC1274: user identifier'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{256} )
```

```
attributetype ( 0.9.2342.19200300.100.1.3
        NAME ( 'mail' 'rfc822Mailbox' )
        DESC 'RFC1274: RFC822 Mailbox'
        EQUALITY caseIgnoreIA5Match
        SUBSTR caseIgnoreIA5SubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{256} )
```

Riguardo le altre direttive globali ci soffermiamo solo sul livello di log a riga 29 (le altre dovrebbero essere abbastanza chiare ed in ogni caso si può fare riferimento ai commenti nel file e al manuale in linea di '**slapd.conf**').

I valori possibili del livello di log sono -1, 0 o potenze del due; di seguito viene riportata la tabella dei valori che si trova nel manuale in linea di '**slapd.conf**':

Livello di log	Descrizione
-1	enable all debugging
0	no debugging
1	trace function calls
2	debug packet handling
4	heavy trace debugging
8	connection management
16	print out packets sent and received
32	search filter processing
64	configuration file processing
128	access control list processing
256	stats log connections/operations/results
512	stats log entries sent
1024	print communication with shell backends
2048	print entry parsing debugging

Il valore inserito può essere tale da attivare più opzioni di log; ad esempio se si sceglie 255 si attivano le opzioni corrispondenti ai livelli 1, 2, 4, 8, 16, 32, 64, 128.

Nella sezione delle specifiche del *backend* notiamo che, alla riga 45, è indicata BDB come base di dati; notiamo inoltre che sarebbe possibile avere più di un *backend* (righe commentate da 49 a 52).

Molto più corposa è la sezione delle specifiche della base di dati, all'interno della quale notiamo subito come sia possibile gestire più basi di dati o alberi con un solo servente OpenLDAP (righe commentate da 125 a 131).

Nel nostro esempio è sufficiente gestire un solo albero e le relative direttive sono comprese tra riga 55 e riga 123.

Una delle più importanti è naturalmente l'indicazione della radice (o suffisso) a riga 61; anche in questo caso il valore corrisponde a quanto indicato al momento della configurazione del pacchetto '**slapd**'.

Alla riga 64 viene indicato dove risiede fisicamente l'archivio; seguono direttive sulla memoria *cache* il bloccaggio degli oggetti, l'indicizzazione dell'archivio, che qui non vengono approfondite.

Nell'ultima parte della sezione ci sono le direttive per le *access list* che servono a indicare chi può accedere all'elenco e quali operazioni può fare.

Nello specifico: con le righe da 95 a 99 si indica che l'utente amministratore può cambiare le parole d'ordine di tutti, che ognuno può cambiare la propria, che gli altri non possono né leggere né scrivere la parola d'ordine di qualcuno.

La seconda direttiva '**`access`**' di riga 110, è presente per evitare malfunzionamenti se si utilizza il meccanismo di autenticazione SASL (*Simple Authentication and Security Layer*); nel nostro caso non servirebbe.

Con la terza (righe da 114 a 116) si stabilisce che l'amministratore può accedere in lettura e scrittura a qualsiasi oggetto, tutti gli altri utenti invece solo in lettura.

Prima di concludere la configurazione del servente, può essere opportuno definire i diritti di accesso di un altro utente, diverso dall'amministratore, da usare per la consultazione dell'elenco in sede di accreditamento.

A questo utente viene concesso solo il diritto di leggere le informazioni relative alle parole d'ordine (cosa sufficiente per le operazioni di accreditamento).

Un'altra possibilità sarebbe quella di accedere all'elenco come amministratore ma qui non viene presa in considerazione.

Entrambe le scelte hano infatti un «difetto»:

- accedendo come utente «normale» si rende impossibile la modifica delle parole d'ordine degli utenti da parte dell'utente ***root*** lavorando dai clienti della rete;
- d'altra parte l'accesso come amministratore comporta la necessità di scrivere in chiaro in alcuni file di configurazione sui client (vedi paragrafo 9.3) la parola d'ordine di tale utente; tali file saranno ovviamente leggibili solo da ***root*** ma la possibilità che tale utenza, per un cliente della rete, venga compromessa è comunque non trascurabile e comporta il rischio che la parola d'ordine dell'amministratore di OpenLDAP cada nelle mani di qualche malintenzionato.

Per motivi di sicurezza è da ritenere preferibile la prima alternativa.

La riga da aggiungere in '`/etc/ldap/slapd.conf`' (fra le righe 96 e 97 già presenti) è la seguente:

```
by dn="cn=auth,dc=max,dc=planck" read
```

avendo evidentemente scelto ***auth*** come nome dell'utente da usare per la consultazione dell'elenco.

Per usare il vecchio file di configurazione '`/etc/ldap/slapd.conf`' occorre convertirne il contenuto nel nuovo formato tramite il seguente procedimento:

- creare il proprio file di configurazione con il vecchio formato e denominarlo '`/etc/ldap/slapd.conf`' (ovviamente la scelta di tale percorso/nome non è obbligatoria);
- creare eventualmente la directory che deve contenere i nuovi archivi di configurazione; ad esempio '`/etc/ldap/slapd.d`';
- eseguire solo la prima delle tre operazioni elencate nel paragrafo 9.2.2 (quella che riguarda l'inserimento della radice dell'elenco, del tipo di archivio ecc.);
- se attivo, fermare il servizio ldap;

- posizionarsi nella directory '`/etc/ldap`' e convertire il file di configurazione con il comando seguente:

  ```
  # slaptest -f slapd.conf -F slapd.d
  ```

- a questo punto può essere necessario cambiare proprietario e gruppo proprietario di tutti i file creati con il comando:

  ```
  # chown -R openldap:openldap slapd.d
  ```

- riavviare il servizio ldap.

9.2.4 Avvio e chiusura di OpenLDAP

Per avviare o riavviare il demone di OpenLDAP, nella Debian o simili, si esegue:

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

oppure:

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

Per fermarlo si esegue:

```
# /etc/init.d/slapd stop
```

Il servizio risponde sulla porta 389 (valore che può essere cambiato intervenendo nel file di configurazione).

Naturalmente è consigliabile fare in modo che il demone '**slapd**' sia inserito fra quelli che vengono avviati automaticamente all'accensione del sistema.

Di solito a questo provvedono gli script di installazione del pacchetto; se ciò non avviene si può ovviare creando un collegamento simbolico nella directory '`/etc/rc2.d`' allo script di attivazione del demone con il comando:

```
# ln -s /etc/init.d/slapd /etc/rc2.d/S19slapd
```

Si presti attenzione al fatto che questo procedimento è valido solo per Debian e distribuzioni derivate, in altri casi varia leggermente.

9.2.5 Popolare l'elenco

Per inserire i dati all'interno dell'elenco e per poi poterli gestire si utilizzano una serie di programmi presenti nei pacchetti '**slapd**' e '**ldap-utils**'.

I primi sono eseguibili solo dall'utente ***root*** e il loro nome inizia con «slap»; gli altri sono eseguibili da tutti gli utenti e il loro nome inizia con «ldap».

Quello che segue è un elenco dei più importanti tra questi programmi.

Di quelli che utilizziamo per caricare e gestire il nostro elenco viene dato tra breve qualche dettaglio di utilizzo; per una panoramica più completa si rimanda, come sempre, ai relativi manuali in linea:

- '**ldapadd**': per inserire oggetti nell'elenco;
- '**ldapdelete**': per cancellare oggetti dall'elenco;

- '**ldapmodify**': per modificare oggetti dell'elenco;
- '**ldappasswd**': per modificare le parole d'ordine ;
- '**ldapsearch**': per cercare dati all'interno dell'elenco e visualizzarli;
- '**slapacl**': per verificare i permessi di accesso agli oggetti;
- '**slapadd**': per inserire oggetti nell'elenco;
- '**slapcat**': per visualizzare il contenuto di tutto l'archivio;
- '**slapdn**': per controllare che una certa stringa, passata come parametro, sia un DN valido per il nostro elenco;
- '**slapindex**': per generare o rigenerare gli indici dell'archivio;
- '**slappasswd**': per generare parole d'ordine cifrate da inserire in archivio (ad esempio con '**ldapmodify**');
- '**slaptest**': per controllare che il contenuto di '**slapd.conf**' sia corretto.

Per inserire i dati nell'elenco si utilizzano file scritti in uno speciale formato chiamato LDIF (*Lightweight Directory Interchange Format*).

Questo formato permette di descrivere gli oggetti dell'elenco e rende possibile il caricamento e lo scaricamento degli stessi in OpenLDAP.

Il formato LDIF è semplicemente un testo con una serie di coppie attributo/valore come nell'esempio che segue relativo ad un ipotetico oggetto «utente GNU/Linux»:

```
dn: uid=fulvio,ou=People,dc=max,dc=planck
uid: fulvio
cn:: ZnVsdmlv
objectClass: account
objectClass: posixAccount
objectClass: top
objectClass: shadowAccount
userPassword: {crypt}$1$HvG0n.P3$/0rbqUdi40yALQEXWunv50
shadowLastChange: 13220
shadowMax: 99999
shadowWarning: 7
loginShell: /bin/bash
uidNumber: 1000
gidNumber: 1000
homeDirectory: /home/fulvio
gecos: fulvio,,,
```

Prima di tutto dobbiamo inserire nell'elenco l'utente ***auth*** da utilizzare per la consultazione dei dati in sede di accreditamento degli utenti della rete, come spiegato alla fine del paragrafo 9.2.2.

Per questo prepariamo un file '`auth.ldif`' come il seguente:

```
dn: cn=auth,dc=max,dc=planck
cn: auth
sn: auth
objectClass: top
objectClass: person
```

```
userPassword: {CRYPT}Eqz.Ufi0SLb3k
```

La parola d'ordine può essere ottenuta con il comando:

```
# slappasswd -h {crypt}
```

e poi incollata nel file.

Quindi fermiamo il servizio OpenLDAP:

```
# /etc/init.d/slapd stop
```

e inseriamo l'utente eseguendo:

```
# slapadd -l auth.ldif
```

In teoria per inserire tutti i dati nell'elenco si dovrebbe operare in modo simile:

- creare uno o più file in formato LDIF (e solitamente con estensione «ldif») contenenti gli oggetti (anche più oggetti per file) da inserire;
- eseguire più volte il comando:

  ```
  # slapadd -l nome_file.ldif
  ```

  passando ogni volta uno dei file ldif creati (l'opzione '**-l**' indica al programma di leggere l'input dal file invece che dallo standard input).

Anche se tecnicamente fattibile, questo procedimento sarebbe senza dubbio lungo e noioso; ecco quindi che sono stati creati degli strumenti automatici di migrazione scritti in Perl che permettono di ottenere automaticamente i file LDIF relativi a molti degli archivi gestiti in un sistema GNU/Linux come ad esempio gli utenti e i gruppi (attingendo i dati dai file '`/etc/passwd`', '`/etc/shadow`', '`/etc/group`').

Questi strumenti si trovano nel pacchetto '**migrationtools**' che è stato già citato come molto utile e quindi senz'altro da installare.

In particolare usiamo (nell'ordine in cui sono elencati):

- '**migrate_base.pl**': per la definizione degli oggetti di base dell'elenco;
- '**migrate_passwd.pl**': per la migrazione degli utenti;
- '**migrate_group.pl**': per la migrazione dei gruppi.

Tutti presenti nella directory '`/usr/share/migrationtools/`'.

Prima di usare questi strumenti occorre modificare il file '`/usr/share/perl5/migrate_common.ph`', contenente le impostazioni di base per la migrazione, in modo che le righe riguardanti il nome del dominio e il suffisso assumano il seguente aspetto:

```
# Default DNS domain
$DEFAULT_MAIL_DOMAIN = "max.planck";

# Default base
$DEFAULT_BASE = "dc=max,dc=planck";
```

Procediamo poi alla creazione dei file LDIF posizionandoci in '`/usr/share/migrationtools/`' ed eseguendo:

```
# ./migrate_base.pl >base.ldif

# ./migrate_passwd.pl /etc/passwd >utenti.ldif

# ./migrate_group.pl /etc/group >gruppi.ldif
```

A questo punto può essere necessario intervenire sui file ottenuti per qualche modifica: ad esempio possiamo togliere dai file '`utenti.ldif`' e '`gruppi.ldif`' gli utenti e i gruppi «di sistema» (come ***bin***, ***mail***, ***news*** ecc.) che non sono utenti e gruppi effettivi.

Sicuramente dobbiamo eliminare dal file '`base.ldif`' le righe seguenti:

```
dn: dc=max,dc=planck
dc: max
objectClass: top
objectClass: domain
```

perché sono relative ad un oggetto già presente nell'elenco (è stato inserito quando si è configurato il pacchetto '**`slapd`**').

Iniziamo quindi a popolare il nostro elenco con:

```
# slapadd -l base.ldif
```

In questo modo vengono inseriti molti oggetti utili all'inserimento successivo di utenti e gruppi.

Se adesso eseguiamo il comando:

```
# slapcat
```

otteniamo una risposta molto più corposa rispetto a quando l'elenco era quasi vuoto; ci sono infatti, oltre alla radice e all'utente amministratore, anche tutti gli oggetti seguenti:

```
dn: ou=Hosts,dc=max,dc=planck
ou: Hosts
objectClass: top
objectClass: organizationalUnit
structuralObjectClass: organizationalUnit
entryUUID: 1cade61c-c183-102a-94fb-6f2027449ad2
creatorsName:
modifiersName:
createTimestamp: 20060816145616Z
modifyTimestamp: 20060816145616Z
entryCSN: 20060816145616Z#000000#00#000000

dn: ou=Rpc,dc=max,dc=planck
ou: Rpc
objectClass: top
objectClass: organizationalUnit
structuralObjectClass: organizationalUnit
entryUUID: 1cae1e20-c183-102a-94fc-6f2027449ad2
creatorsName:
modifiersName:
createTimestamp: 20060816145616Z
```

```
modifyTimestamp: 20060816145616Z
entryCSN: 20060816145616Z#000001#00#000000

dn: ou=Services,dc=max,dc=planck
ou: Services
objectClass: top
objectClass: organizationalUnit
structuralObjectClass: organizationalUnit
entryUUID: 1cb07e90-c183-102a-94fd-6f2027449ad2
creatorsName:
modifiersName:
createTimestamp: 20060816145616Z
modifyTimestamp: 20060816145616Z
entryCSN: 20060816145616Z#000002#00#000000

dn: nisMapName=netgroup.byuser,dc=max,dc=planck
nisMapName: netgroup.byuser
objectClass: top
objectClass: nisMap
structuralObjectClass: nisMap
entryUUID: 1cb08fa2-c183-102a-94fe-6f2027449ad2
creatorsName:
modifiersName:
createTimestamp: 20060816145616Z
modifyTimestamp: 20060816145616Z
entryCSN: 20060816145616Z#000003#00#000000

dn: ou=Mounts,dc=max,dc=planck
ou: Mounts
objectClass: top
objectClass: organizationalUnit
structuralObjectClass: organizationalUnit
entryUUID: 1cb0acc6-c183-102a-94ff-6f2027449ad2
creatorsName:
modifiersName:
createTimestamp: 20060816145616Z
modifyTimestamp: 20060816145616Z
entryCSN: 20060816145616Z#000004#00#000000

dn: ou=Networks,dc=max,dc=planck
ou: Networks
objectClass: top
objectClass: organizationalUnit
structuralObjectClass: organizationalUnit
entryUUID: 1cb0ba9a-c183-102a-9500-6f2027449ad2
creatorsName:
modifiersName:
createTimestamp: 20060816145616Z
modifyTimestamp: 20060816145616Z
entryCSN: 20060816145616Z#000005#00#000000

dn: ou=People,dc=max,dc=planck
ou: People
objectClass: top
```

```
objectClass: organizationalUnit
structuralObjectClass: organizationalUnit
entryUUID: 1cb0c832-c183-102a-9501-6f2027449ad2
creatorsName:
modifiersName:
createTimestamp: 20060816145616Z
modifyTimestamp: 20060816145616Z
entryCSN: 20060816145616Z#000006#00#000000

dn: ou=Group,dc=max,dc=planck
ou: Group
objectClass: top
objectClass: organizationalUnit
structuralObjectClass: organizationalUnit
entryUUID: 1cb0d5a2-c183-102a-9502-6f2027449ad2
creatorsName:
modifiersName:
createTimestamp: 20060816145616Z
modifyTimestamp: 20060816145616Z
entryCSN: 20060816145616Z#000007#00#000000

dn: ou=Netgroup,dc=max,dc=planck
ou: Netgroup
objectClass: top
objectClass: organizationalUnit
structuralObjectClass: organizationalUnit
entryUUID: 1cb1cb4c-c183-102a-9503-6f2027449ad2
creatorsName:
modifiersName:
createTimestamp: 20060816145616Z
modifyTimestamp: 20060816145616Z
entryCSN: 20060816145616Z#000008#00#000000

dn: ou=Protocols,dc=max,dc=planck
ou: Protocols
objectClass: top
objectClass: organizationalUnit
structuralObjectClass: organizationalUnit
entryUUID: 1cb1dc90-c183-102a-9504-6f2027449ad2
creatorsName:
modifiersName:
createTimestamp: 20060816145616Z
modifyTimestamp: 20060816145616Z
entryCSN: 20060816145616Z#000009#00#000000

dn: ou=Aliases,dc=max,dc=planck
ou: Aliases
objectClass: top
objectClass: organizationalUnit
structuralObjectClass: organizationalUnit
entryUUID: 1cb1ea82-c183-102a-9505-6f2027449ad2
creatorsName:
modifiersName:
createTimestamp: 20060816145616Z
```

```
modifyTimestamp: 20060816145616Z
entryCSN: 20060816145616Z#00000a#00#000000

dn: nisMapName=netgroup.byhost,dc=max,dc=planck
nisMapName: netgroup.byhost
objectClass: top
objectClass: nisMap
structuralObjectClass: nisMap
entryUUID: 1cb201d4-c183-102a-9506-6f2027449ad2
creatorsName:
modifiersName:
createTimestamp: 20060816145616Z
modifyTimestamp: 20060816145616Z
entryCSN: 20060816145616Z#00000b#00#000000
```

Ora non rimane che inserire utenti e gruppi con i comandi:

```
# slapadd -l utenti.ldif

# slapadd -l gruppi.ldif
```

Si può verificare con '**slapcat**' che tutti i dati siano stati inseriti.

Ecco una parte della risposta, riguardante gli oggetti nell'elenco corrispondenti all'utente e al gruppo ***rossimario***:

```
dn: uid=rossimario,ou=People,dc=max,dc=planck
uid: rossimario
cn: Rossi Mario
objectClass: account
objectClass: posixAccount
objectClass: top
objectClass: shadowAccount
userPassword:: e2NyeXB0fSQxJEZjRlJNWnRrJG9YZWZmTldMd1RYSnNmV281amVtNTE=
shadowLastChange: 13376
shadowMax: 99999
shadowWarning: 7
loginShell: /bin/bash
uidNumber: 1001
gidNumber: 1001
homeDirectory: /home/rossimario
gecos: Rossi Mario,,0422-1234567,
structuralObjectClass: account
entryUUID: 2355333c-c19a-102a-836a-21b74a8ee554
creatorsName:
modifiersName:
createTimestamp: 20060816174105Z
modifyTimestamp: 20060816174105Z
entryCSN: 20060816174105Z#000000#00#000000

dn: cn=rossimario,ou=Group,dc=max,dc=planck
objectClass: posixGroup
objectClass: top
cn: rossimario
userPassword:: e2NyeXB0fXg=
```

```
gidNumber: 1001
structuralObjectClass: posixGroup
entryUUID: 25f2f44e-c19a-102a-8997-638e7699ad6b
creatorsName:
modifiersName:
createTimestamp: 20060816174110Z
modifyTimestamp: 20060816174110Z
entryCSN: 20060816174110Z#000000#00#000000
```

Per continuare a utilizzare il servente OpenLDAP, ricordiamo poi di riattivarlo:

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

9.2.6 Gestire i dati dell'elenco

Vediamo brevemente come è possibile cancellare e modificare i dati dell'elenco e come si possono fare le ricerche; le prime due operazioni possono essere svolte solo da utenti autorizzati (per come abbiamo configurato il servente, solo dall'utente amministratore).

Vediamo poi anche un modo alternativo a '**slapadd**' per inserire i dati.

Dei comandi illustrati in questo paragrafo vengono fornite solo le linee essenziali di utilizzo; per i dettagli sulle numerose opzioni previste si rimanda come al solito ai manuali in linea.

Prima di utilizzare i programmi di gestione dell'elenco occorre modificare il file di configurazione '`/etc/ldap/ldap.conf`' (o '`/etc/ldap.conf`') in modo che contenga le informazioni mostrate di seguito (il contenuto preciso del file varia secondo la versione dei pacchetti che si usano):

```
# LDAP Defaults
#

# See ldap.conf(5) for details
# This file should be world readable but not world writable.

BASE    dc=max, dc=planck
URI     ldap://127.0.0.1:389
```

Questo è necessario anche nelle macchine clienti nelle quali, evidentemente, l'indirizzo indicato nella riga '**URI**' deve essere quello del servente e non quello del ***localhost***; in caso ci fossero problemi di collegamento al servente si può provare ad inserire la riga '**host numero_ip**' al posto di '**URI**'.

La presenza di queste impostazioni permette di usare i comandi di manipolazione dell'elenco senza indicare ogni volta informazioni come la radice dell'albero, l'indirizzo e la porta su cui è in ascolto il servente.

Per ulteriori dettagli sul contenuto di questo file si veda il manuale in linea di '**ldap.conf**'.

9.2.6.1 Cancellazione dei dati

Il comando per cancellare un oggetto è '**ldapdelete**'; vediamo subito un esempio supponendo di voler cancellare l'utente ***rossimario***:

```
# ldapdelete -x -D "cn=admin,dc=max,dc=planck" ↩
↪ "uid=rossimario,ou=People,dc=max,dc=planck" -W
```

Il significato delle opzioni e degli argomenti è questo:

- '**-D "cn=admin,dc=max,dc=planck"**': l'utente che si accredita per fare l'operazione è ***admin***;
- '**-x**': viene usato l'accreditamento semplice e non SASL;
- '**-W**': la parola d'ordine di ***admin*** viene richiesta in modo interattivo;
- '**"uid=rossimario,ou=People,dc=max,dc=planck"**': è il DN dell'oggetto da cancellare.

Alcune altre opzioni importanti:

- '**-w psw**': al posto di '**-W**', permette di indicare direttamente la parola d'ordine;
- '**-y nome_file**': al posto di '**-W**', permette di indicare un file contenente la parola d'ordine;
- '**-f nome_file**': al posto dell'indicazione del DN, permette di inserire una lista di oggetti da cancellare in un file (un DN per riga);
- '**-H ldap://nome_host:num_porta**': la richiesta di cancellazione è fatta al nodo ***nome_host*** sulla porta ***num_porta*** (i valori per difetto sono rispettivamente ***localhost*** e ***389***).

9.2.6.2 Modifica dei dati

Per la modifica si usa il comando '**ldapmodify**'; anche in questo caso esaminiamo un esempio:

```
# ldapmodify -x -D "cn=admin,dc=max,dc=planck" -f file_modifiche.txt -W
```

Alcune opzioni (compresa '**-H**') hanno esattamente lo stesso significato visto nel comando precedente; in questo caso si passa in input un ***file_modifiche*** che contiene le istruzioni sulle modifiche da apportare.

Il contenuto del file potrebbe essere simile al seguente:

```
dn: uid=rossimario,ou=People,dc=max,dc=planck
changetype: modify
replace: homeDirectory
homeDirectory:/home/rossi
-
replace: loginShell
loginShell: /bin/zsh
-
```

Il file contiene uno o più DN da modificare e per ognuno il tipo di modifica da fare (nell'esempio è ***modify*** ma potrebbe anche essere ***add*** o ***delete*** in modo da rimpiazzare rispettivamente i comandi '**ldapadd**' o '**ldapdelete**'). Qui si chiede di modificare gli attributi ***homedirectory*** e ***loginShell*** con i nuovi valori indicati; è anche possibile eliminare o aggiungere attributi (se consentito dallo schema) rispettivamente scrivendo ***delete*** o ***add*** al posto di ***replace***.

Una volta eseguito il comando, ecco il nuovo aspetto dell'oggetto ***mariorossi*** ottenuto con '**slapcat**':

```
dn: uid=rossimario,ou=People,dc=max,dc=planck
uid: rossimario
cn: Rossi Mario
objectClass: account
objectClass: posixAccount
objectClass: top
objectClass: shadowAccount
userPassword:: e2NyeXB0fSQxJEZjRlJNWnRrJG9YZWZmTldMd1RYSnNmV281amVtNTE=
shadowLastChange: 13376
shadowMax: 99999
shadowWarning: 7
uidNumber: 1001
gidNumber: 1001
gecos: Rossi Mario,,0422-1234567,
structuralObjectClass: account
entryUUID: 02a4028a-c1b8-102a-8b59-6d8a33f08c2a
creatorsName:
createTimestamp: 20060816211455Z
homeDirectory: /home/rossi
loginShell: /bin/zsh
entryCSN: 20060816212358Z#000000#00#000000
modifiersName: cn=admin,dc=max,dc=planck
modifyTimestamp: 20060816212358Z
```

Esiste anche la possibilità di utilizzare '**ldapmodify**' senza passare il file di comandi; in questo caso si inseriscono le istruzioni di modifica da tastiera, con la stessa sintassi usata nel file, concludendo l'input con [*Ctrl d*].

9.2.6.3 Aggiunta di dati

In precedenza abbiamo utilizzato il comando '**slapadd**' per inserire oggetti nell'elenco. La stessa operazione si può fare anche con il comando '**ldapadd**' con la differenza che non occorre essere l'utente ***root*** e che il servente OpenLDAP deve essere attivo.

Il modo migliore per inserire oggetti nell'elenco è quello di preparare dei file LDIF contenenti i dati e passarli in input a '**ldapadd**'.

Nel caso si debbano aggiungere utenti o gruppi si può ricorrere come «modelli» ai file LDIF creati automaticamente dalle procedure di migrazione.

Ecco un esempio di uso di '**ldapadd**':

```
# ldapadd -x -D "cn=admin,dc=max,dc=planck" -f file_dati.ldif -W
```

Non occorre avere i privilegi di ***root*** per eseguire il comando ma occorre comunque accreditarsi (come per la cancellazione e la modifica) presso il servente OpenLDAP come utente abilitato alla scrittura dei dati (in questo caso l'utente ***admin***).

Le opzioni sono le stesse viste per i comandi precedenti e anche qui è prevista l'opzione '**-H**'.

Di seguito vediamo il contenuto di un possibile file di input per l'inserimento di un utente e di uno per il corrispondente gruppo; si tenga presente che si possono comunque inserire più elementi contemporaneamente semplicemente predisponendo i corrispondenti gruppi di righe nel file di input:

```
dn: uid=bianchipaolo,ou=People,dc=max,dc=planck
uid: bianchipaolo
cn: Bianchi Paolo
objectClass: account
objectClass: posixAccount
objectClass: top
objectClass: shadowAccount
userPassword: XXXXXXXXXXXXXXXXXXXX
shadowLastChange: 13376
shadowMax: 99999
shadowWarning: 7
loginShell: /bin/bash
uidNumber: 1002
gidNumber: 1002
homeDirectory: /home/bianchipaolo
gecos: Bianchi Paolo,,0422-1234567,
```

```
dn: cn=bianchipaolo,ou=Group,dc=max,dc=planck
objectClass: posixGroup
objectClass: top
cn: bianchipaolo
userPassword: {crypt}x
gidNumber: 1002
```

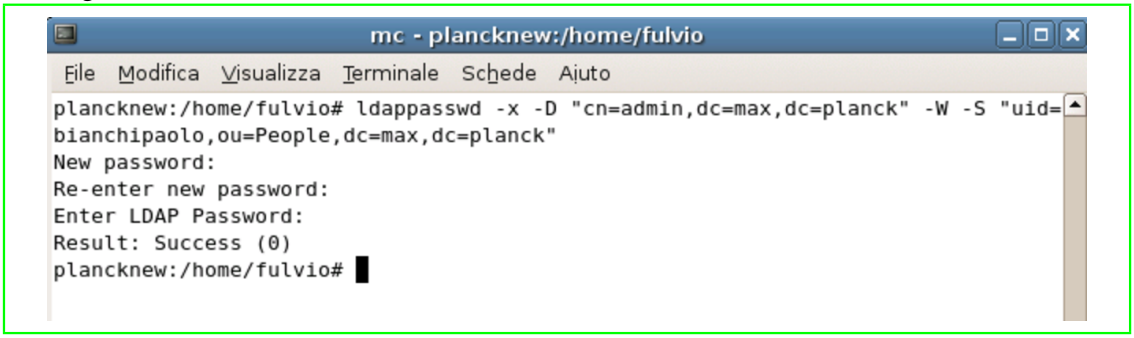
Riguardo alla parola d'ordine si può agire in due modi diversi:

- generarla con il comando '**slappasswd**' (vedere il manuale in linea per i dettagli) e poi copiare la parola d'ordine cifrata ottenuta nel file di input come valore dell'attributo ***userPassword***; ovviamente questo va fatto prima di eseguire il comando di caricamento dei dati;
- aggiornare la parola d'ordine dopo avere inserito l'utente con il comando:

  ```
  # ldappasswd -x -D "cn=admin,dc=max,dc=planck" -W ↩
  ↪-S "uid=bianchipaolo,ou=People,dc=max,dc=planck"
  ```

Usando il secondo metodo si riceve richiesta della nuova parola d'ordine dell'utente e della sua conferma (è l'effetto dell'opzione '**-S**') e poi della parola d'ordine dell'utente (amministratore) che si collega per svolgere l'operazione (vedere figura 9.24).

Figura 9.24.



Anche il comando '**ldappasswd**' prevede la possibilità di interagire con un servente diverso dal ***localhost*** grazie all'opzione '**-H**'.

Una volta che è stata assegnata la prima parola d'ordine ad un utente, questi può cambiarsela in ogni momento con un comando del tipo:

```
# ldappasswd -x -D "uid=bianchipaolo,ou=People,dc=max,dc=planck" -W -S
```

In questo caso non è necessario specificare il DN dell'utente di cui variare la parola d'ordine perché, in mancanza di questa informazione, il sistema agisce sull'utente che si collega al servente (in questo caso ***bianchipaolo***).

Per essere certi che le parole d'ordine vengano gestite correttamente con i comandi di OpenLDAP si deve fare in modo che vengano create con lo stesso metodo di cifratura utilizzato dal sistema GNU/Linux per le parole d'ordine degli utenti migrati in precedenza.
Sulla macchina usata per i nostri esempi tale metodo è '**crypt**' e quindi occorre aggiungere nel file di configurazione '`/etc/ldap/slapd.conf`', fra le direttive globali, questa riga:

```
password-hash {CRYPT}
```

in mancanza di indicazioni viene invece applicato '**SSHA**'.

9.2.6.4 Ricerca di informazioni

Il comando per effettuare ricerche nell'elenco è '**ldapsearch**'.

Le opzioni e le possibilità di utilizzo sono numerose e quindi anche in questo caso si evita l'esame di tutti i dettagli, che sono comunque a disposizione nel manuale in linea, e si commentano alcuni esempi, ricordando che molte opzioni ('**-H**, **-x**, **-D**, **-W**') hanno esattamente lo stesso significato visto nel paragrafo precedente.

```
# ldapsearch -x -LLL -b "dc=max,dc=planck"
```

Si chiede il contenuto di tutto l'elenco che ha come radice ***dc=max,dc=planck*** (opzione '**-b**'); l'opzione '**-L**' serve ad avere il risultato in formato LDIF, se sono due si disabilitano i commenti e se sono tre si disabilita anche la visualizzazione della versione di LDIF.

L'indicazione della base della ricerca può essere evitata se si inserisce nel file di configurazione '`/etc/ldap/slapd.conf`' tra le direttive globali, la riga seguente:

```
defaultsearchbase dc=max,dc=planck
```

Come si vede non è necessario (anche se non è proibito) collegarsi al servente con credenziali valide perché la lettura delle informazioni è libera; per questo motivo nella risposta non appaiono gli attributi sensibili come la ***userPassword***.

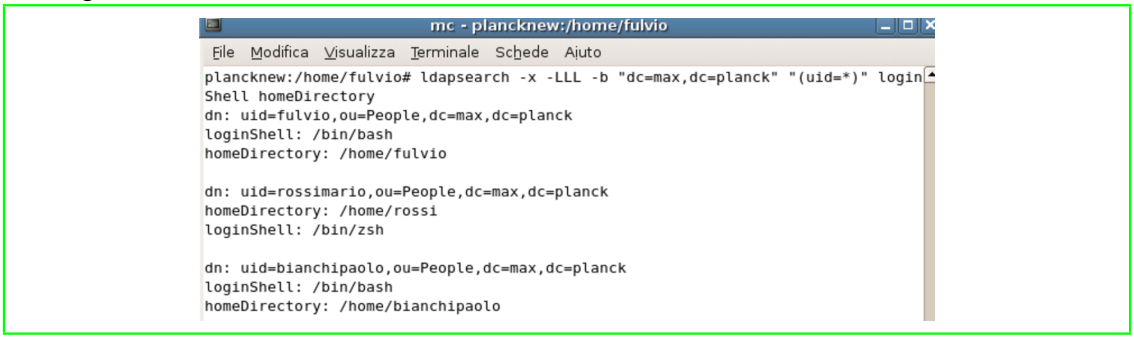
```
# ldapsearch -x -LLL -b "dc=max,dc=planck" "(cn=*)"
```

Si chiedono tutti gli oggetti che hanno un attributo ***cn*** presente, qualsiasi sia il valore di quest'ultimo.

```
# ldapsearch -x -LLL -b "dc=max,dc=planck" "(uid=*)"↩
↪loginShell homeDirectory
```

Si chiedono tutti gli utenti e si vogliono visualizzare solo gli attributi ***loginShell*** e ***homeDirectory***; in figura 9.27 si vede l'effetto del comando.

Figura 9.27.




```
# ldapsearch -x -LLL -b "dc=max,dc=planck" "(ou=p*)"
```

Si chiedono tutte le *organizational unit* il cui nome inizia con «p» o «P» (si ricorda che i valori degli attributi non sono *case sensitive*).

```
# ldapsearch -x -LLL -b "dc=max,dc=planck"↩
↪ "(&(uid=*)(loginshell=/bin/bash))" loginShell homeDirectory
```

In questo ultimo esempio si vede l'uso di un filtro di ricerca multiplo: si vogliono gli oggetti con attributo ***uid*** presente e ***loginShell*** uguale a ***/bin/bash***; si noti la sintassi che prevede l'indicazione dell'operatore booleano in testa alla stringa di ricerca.

Gli operatori booleani sono:

- '`&`': and;
- '`|`': or;
- '`!`': not.

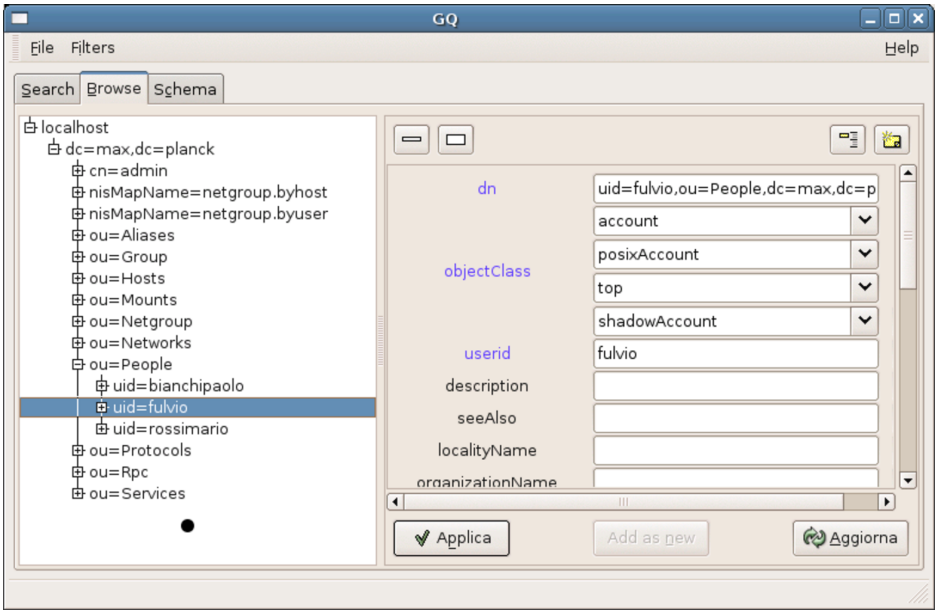
9.2.7 Clienti grafici per OpenLDAP

Esistono vari strumenti grafici che facilitano la gestione di un elenco OpenLDAP.

Per i motivi già illustrati nella premessa, è consigliabile utilizzare questo tipo di programmi solo quando si è acquisita sufficiente conoscenza sulla struttura, la configurazione, la gestione di un elenco OpenLDAP, in modo da sfruttarne la comodità e velocità di utilizzo sapendo però quello che si sta facendo.

A titolo di esempio citiamo gq [2] (⟨*http://gq-project.org/*⟩) del quale vediamo un immagine in figura 9.28.

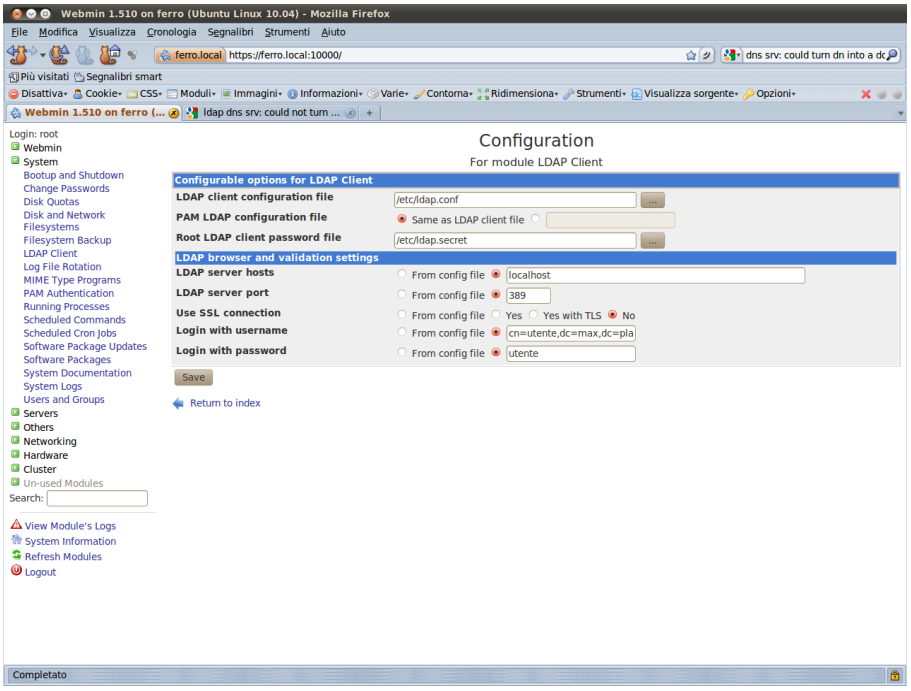
Figura 9.28.



e l'ottimo programma webmin [3] (⟨*http://www.webmin.com*⟩) che permette la gestione completa di un sistema GNU/Linux con una comoda interfaccia Web (alla porta 10000) e che possiede anche moduli per gli utenti OpenLDAP ('**`ldap users and groups`**' e '**`ldap client`**' utilizzabili in caso si siano installati anche i pacchetti di OpenLDAP per l'autenticazione degli utenti di cui parliamo nel paragrafo 9.3).

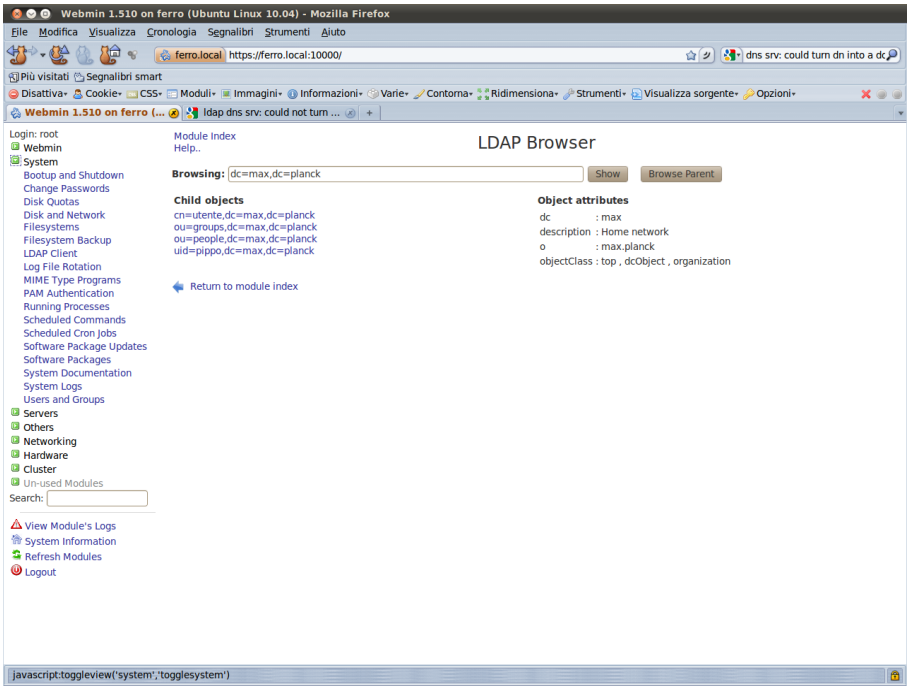
Nella figura 9.29 vediamo la schermata di configurazione del modulo '**`ldap client`**':

Figura 9.29.



In figura 9.30 vediamo invece una delle schermate, dello stesso modulo, per l'interrogazione dei dati dell'elenco OpenLDAP.

Figura 9.30.



9.3 Autenticazione degli utenti di rete OpenLDAP

Una volta installato e configurato il servente OpenLDAP si può passare a configurare i clienti affinché lo utilizzino per l'accreditamento degli utenti; a questo proposito possiamo senz'altro considerare tra i clienti anche la macchina dove è installato OpenLDAP.

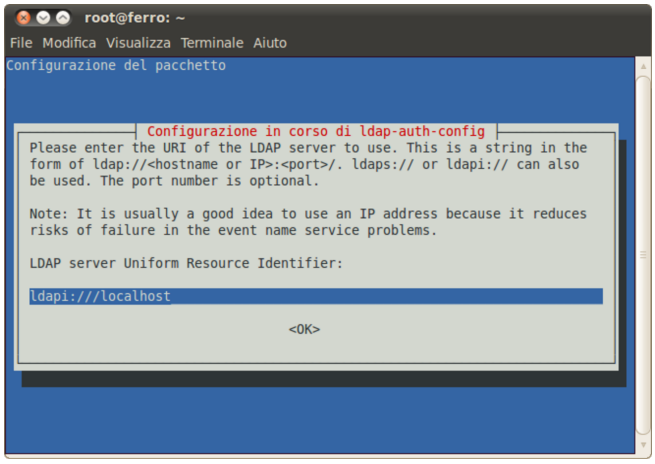
9.3.1 Configurazione dei pacchetti necessari

Devono essere installati i pacchetti '**`nscd`**' (che non richiede configurazioni particolari), '**`ldap-auth-client`**', '**`ldap-auth-config`**', '**`auth-client-config`**', '**`libnss-ldap`**' e '**`libpam-ldap`**' per i quali invece la procedura di installazione richiede alcune informazioni.

Si ricevono varie richieste riguardanti la configurazione di '**`ldap-auth-config`**' a partire dall'URI (*Uniform Resource Identifier*) del servente.

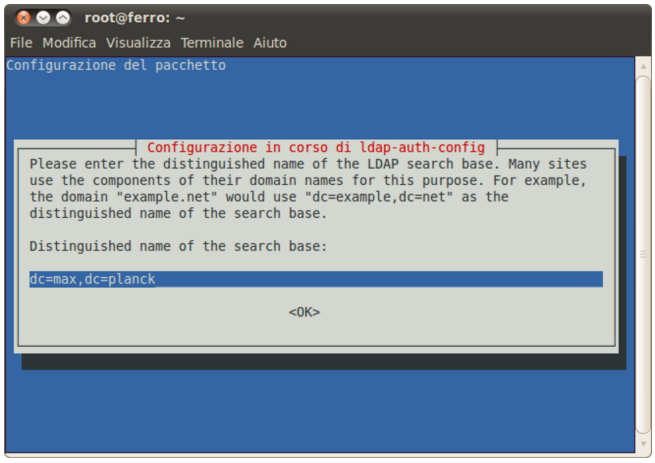
Nella figura 9.31 è visibile la relativa schermata in cui si suppone che il servente sia la macchina su cui si stanno configurando anche gli strumenti «lato cliente»; nella realtà questo è ovviamente il caso meno frequente, quindi consideriamo di dover indicare, al posto di ***localhost***, l'indirizzo di una macchina remota.

Figura 9.31.



Successivamente viene richiesto il suffisso dell'elenco da utilizzare (figura 9.32).

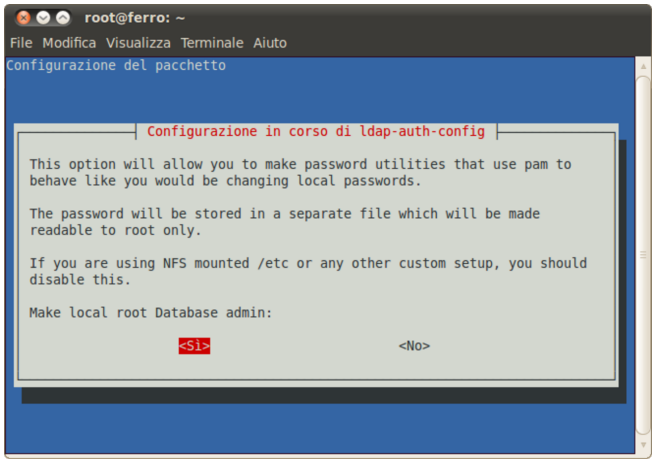
Figura 9.32.



Quindi è la volta della versione di LDAP (scegliere la 3).

A seguire (figura 9.33) viene chiesto se si vuole che le *password utility* che utilizzano '**pam**' si comportino come se si stessero usando le parole d'ordine locali (ripondere affermativamente).

Figura 9.33.



La schermata successiva permette di specificare se l'accesso all'archivio di OpenLDAP richiede

accreditamento (rispondere affermativamente).

Infine si devono immettere i dati dell'utente abilitato a cambiare le parole d'ordine (deve essere un utente privilegiato, vedi figura 9.34) e dell'utente usato per collegarsi all'elenco (meglio che non sia un utente privilegiato vedi figura 9.35).

Quest'ultimo utente ovviamente deve essere presente nell'elenco; per il suo inserimento si faccia riferimento a quanto descritto nel paragrafo 9.2.5.

Figura 9.34.

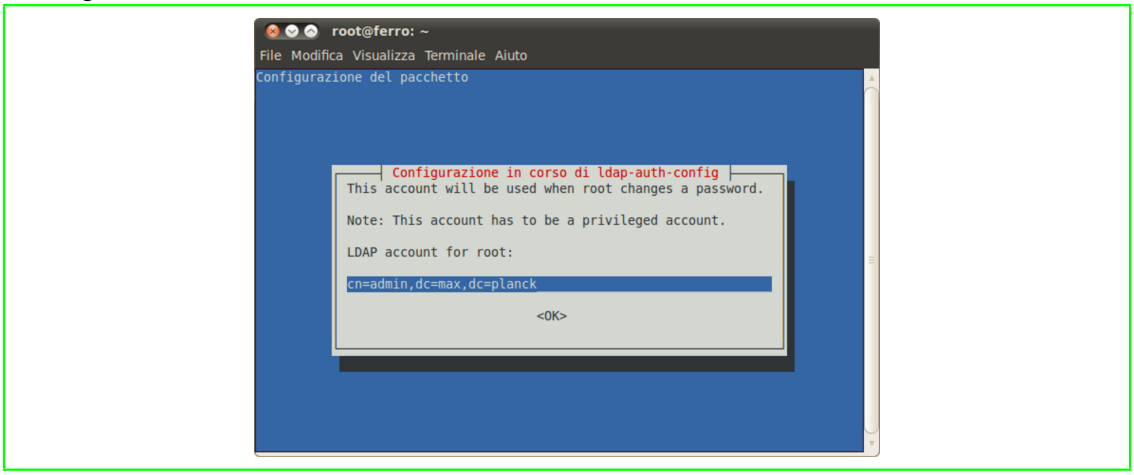
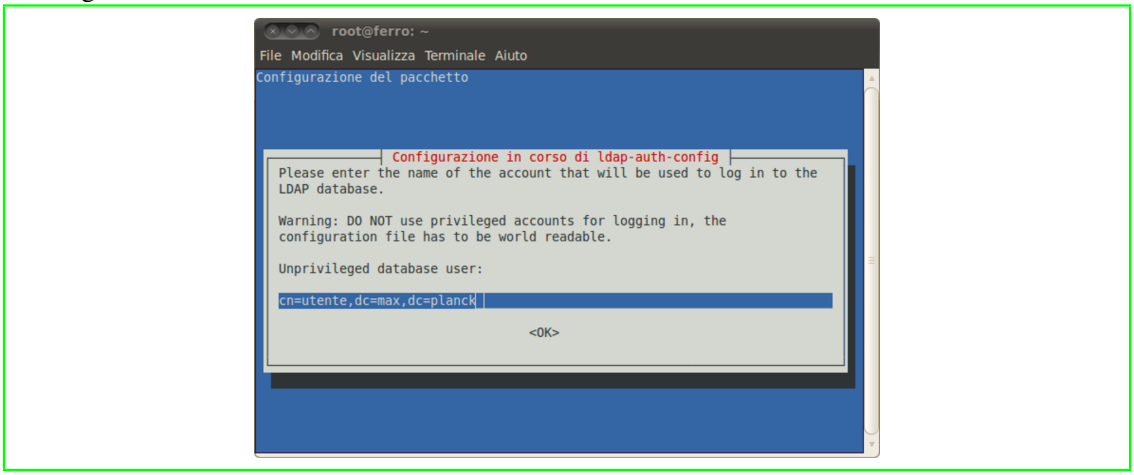


Figura 9.35.



Entrambe le schermate sono seguite dalla richiesta di immissione delle parole d'ordine degli utenti prescelti.

Una volta ultimata questa configurazione occorre riavviare il demone '**nscd**':

```
# /etc/init.d/nscd restart
```

9.3.1.1 Modifiche alla configurazione di NSS

Nel file '`/etc/nsswitch.conf`', contenente la configurazione del servizio NSS, è necessario aggiungere l'archivio gestito con OpenLDAP tra le fonti dei dati relativi a utenti e gruppi come mostrato di seguito:

```
passwd:     compat ldap
shadow:     compat ldap
group:      compat ldap
```

L'ordine con cui vengono elencate le fonti è significativo e quindi è opportuno lasciare la priorità a ***compat*** in modo che per primi siano interrogati i file di sistema.

A questo punto si è già in grado di verificare con il comando:

```
# getent passwd
```

che gli utenti dell'elenco appaiano fra quelli riconosciuti dal sistema.

Prima di fare questa prova è consigliabile togliere gli utenti e i gruppi dai file di configurazione del sistema '`/etc/passwd`', '`/etc/shadow`', '`/etc/group`', cancellando le righe relative o usando i comandi '**`userdel`**' o '**`deluser`**' e '**`groudel`**' o '**`delgroup`**'; in questo modo siamo sicuri che gli utenti e i gruppi elencati sono davvero quelli di OpenLDAP.

Si può inoltre verificare il buon esito dell'accreditamento degli utenti sia sul sistema che ospita il cliente che su macchine clienti in cui si siano effettuate le configurazioni appena esaminate.

Prima di fare questa prova occorre però configurare un altro servizio che è molto utile nel contesto che stiamo esaminando: NFS (*Network File System*), con il quale possiamo centralizzare anche le directory personali degli utenti della rete GNU/Linux.

La macchina sulla quale far risiedere le directory può essere una qualsiasi nella rete, ma ha sicuramente senso che sia la stessa che accredita gli utenti, cioè quella dove è installato OpenLDAP.

Un esame approfondito del servizio NFS esula dagli scopi di queste dispense; ci limitiamo alle nozioni indispensabili il compito che ci proponiamo.

Nella macchina servente si devono installare i pacchetti '**`nfs-kernel-server`**', '**`nfs-common`**' e '**`portmap`**' e si deve configurare il file '`/etc/exports`' come mostrato di seguito:

```
# /etc/exports: the access control list for filesystems which may be exported
#               to NFS clients.  See exports(5).
#
/home           10.0.0.0/255.0.0.0(rw)
```

Il significato della configurazione è abbastanza evidente: si vuole condividere la directory ***/home*** per tutte le macchine della rete con indirizzo ***10.0.0.0*** e maschera di rete ***255.0.0.0*** dando la possibilità di leggere e scrivere tale directory (fatti salvi i permessi sui singoli elementi in essa contenuti).

Si deve poi riavviare il servente NFS:

```
# /etc/init.d/nfs-kernel-server restart
```

Sulle macchine clienti bisogna invece installare i pacchetti '**`nfs-common`**' e '**`portmap`**' e intervenire nel file '`/etc/fstab`' aggiungendo una riga come questa:

```
10.0.0.3:/home    /home    nfs
```

Significa che la directory '`/home`' del servente (che si suppone abbia indirizzo ***10.0.0.3***) viene montata sulla '`/home`' di quella macchina, che deve esistere ed essere possibilmente vuota (il montaggio non cancella il contenuto preesistente ma lo rende irraggiungibile).

Affinché il montaggio avvenga bisogna però anche eseguire:

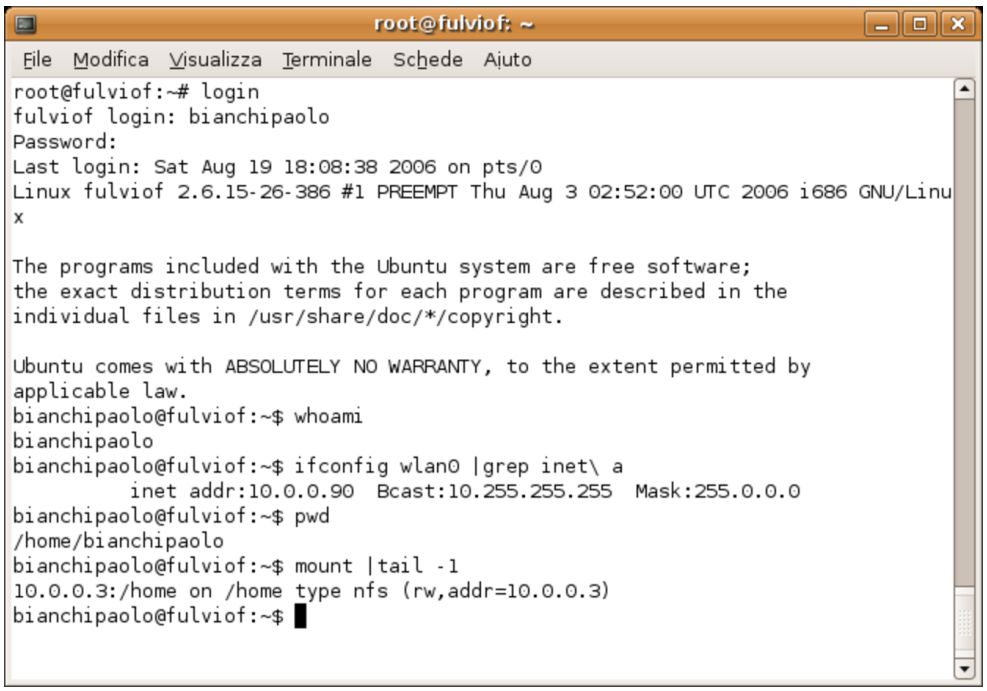
```
# mount -a
```

ma solo la prima volta perché i montaggi indicati nel file '`/etc/fstab`' vengono fatti automaticamente ad ogni riavvio del sistema.

Una volta fatte queste operazioni si può provare a collegarsi su una macchina cliente come utente presente nell'elenco OpenLDAP.

Nella figura 9.39 viene mostrata tale operazione con la successiva esecuzione di vari comandi che permettono di constatare come effettivamente l'utente si sia accreditato da remoto.

Figura 9.39.



9.3.1.2 Modifiche ai file di configurazione dei moduli PAM

Durante la configurazione del pacchetto '**`ldap-auth-config`**' dovrebbe avvenire anche la configurazione del servizio PAM.

In caso contrario si può procedere manualmente ricordando le precauzioni da prendere prima di modificare i moduli PAM che sono state elencate nel paragrafo 7.1.3.

Lo scopo delle modifiche è fare in modo che le applicazioni presenti su una macchina GNU/Linux ('**`login`**', '**`gdm`**', '**`ssh`**' ecc.) effettuino l'autenticazione degli utenti attraverso OpenLDAP:

Nei file di configurazione PAM dei servizi che si vogliono autenticare con OpenLDAP si devono aggiungere righe simili a quelle sotto elencate prima delle analoghe righe che fanno riferimento a '**`pam_unix.so`**' con opzione '**`required`**'.

```
auth     sufficient   pam_ldap.so
account  sufficient   pam_ldap.so
session  sufficient   pam_ldap.so
password sufficient   pam_ldap.so
```

In Debian e distribuzioni derivate si può intervenire nei seguenti file (usati da tutte le applicazioni):

- '`/etc/pam.d/common-account`'
- '`/etc/pam.d/common-auth`'
- '`/etc/pam.d/common-password`'
- '`/etc/pam.d/common-session`'

Anche in questo caso è possibile creare automaticamente la directory personale di un utente al suo primo accreditamento (come visto nel paragrafo 7.1.3) aggiungendo in testa al file '`/etc/pam.d/common-session`' la seguente riga:

```
session   required   /lib/security/pam_mkhomedir.so  skel=/etc/skel/  umask=0022
```

9.4 Usare Samba con OpenLDAP

Vediamo adesso come utilizzare Samba come PDC con archivio utenti gestito da OpenLDAP.

Questa possibilità è presente stabilmente dalla versione 3 di Samba; con le versioni precedenti era necessario configurare e compilare manualmente il software.

Si noti inoltre che, dovendo aggiornare un servente Samba dalla versione 2.2 alla 3 si deve prevedere una conversione degli oggetti dell'elenco in quanto non è stata mantenuta la compatibilità tra gli schemi di elenco utilizzati e si è passati dal tipo di oggetto '**`sambaAccount`**' a '**`sambaSamAccount`**'.

Per la conversione è possibile utilizzare lo script '`convertSambaAccount`' disponibile (in Debian) in '`/usr/share/doc/samba-doc/examples/LDAP/`'.

9.4.1 Creazione dell'elenco per Samba

Supponiamo di avere già un archivio di utenti GNU/Linux e che l'elenco OpenLDAP sia inizialmente vuoto; quindi ripartiamo da zero eliminando gli eventuali oggetti inseriti con le prove precedenti.

Il modo più rapido per ottenere questo è fermare il servente '**`slapd`**', cancellare il contenuto della directory '`/var/lib/ldap/`', e ricreare l'elenco nuovo eseguendo il comando:

```
# dpkg-reconfigure slapd
```

Si devono poi reinserire schemi e informazioni di base e configurare '**`ldap-auth-config`**' come descritto nei paragrafi 9.2.2 e 9.3.1.

Dobbiamo poi installare il pacchetto '**`smbldap-tools`**' che contiene tutto quello che serve per i nostri scopi; i programmi presenti sono scritti in Perl e sono i seguenti (per i dettagli di funzionamento si rimanda alla consultazione dei rispettivi manuali in linea):

- /usr/sbin/smbldap-groupadd
- /usr/sbin/smbldap-groupdel
- /usr/sbin/smbldap-groupmod

- /usr/sbin/smbldap-groupshow
- /usr/sbin/smbldap-passwd
- /usr/sbin/smbldap-populate
- /usr/sbin/smbldap-useradd
- /usr/sbin/smbldap-userdel
- /usr/sbin/smbldap-userinfo
- /usr/sbin/smbldap-usermod
- /usr/sbin/smbldap-usershow

Prima di iniziare a definire l'elenco occorre modificare la configurazione di questo pacchetto e di OpenLDAP compiendo le seguenti operazioni:

1. copiare i file '`smbldap.conf`' e '`smbldap_bind.conf`' da '`/usr/share/doc/smbldap-tools/examples/`' a '`/etc/smbldap-tools/`' con i comandi:

   ```
   # zcat /usr/share/doc/smbldap-tools/examples/smbldap.conf.gz >↩
   ↪/etc/smbldap-tools/smbldap.conf

   # cp /usr/share/doc/smbldap-tools/examples/smbldap_bind.conf↩
   ↪/etc/smbldap-tools/
   ```

2. modificare il file '`smbldap.conf`' come mostrato di seguito:

```
1  # $Source: /opt/cvs/samba/smbldap-tools/smbldap.conf,v $
2  # $Id: smbldap.conf,v 1.18 2005/05/27 14:28:47 jtournier Exp $
3  #
4  # smbldap-tools.conf : Q & D configuration file for smbldap-tools
5
6  #  This code was developped by IDEALX (http://IDEALX.org/) and
7  #  contributors (their names can be found in the CONTRIBUTORS file).
8  #
9  #                 Copyright (C) 2001-2002 IDEALX
10 #
11 #  This program is free software; you can redistribute it and/or
12 #  modify it under the terms of the GNU General Public License
13 #  as published by the Free Software Foundation; either version 2
14 #  of the License, or (at your option) any later version.
15 #
16 #  This program is distributed in the hope that it will be useful,
17 #  but WITHOUT ANY WARRANTY; without even the implied warranty of
18 #  MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
19 #  GNU General Public License for more details.
20 #
21 #  You should have received a copy of the GNU General Public License
22 #  along with this program; if not, write to the Free Software
23 #  Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307,
24 #  USA.
25
26 #  Purpose :
27 #       . be the configuration file for all smbldap-tools scripts
28
```

```
29  ##############################################################################
30  #
31  # General Configuration
32  #
33  ##############################################################################
34
35  # Put your own SID. To obtain this number do: "net getlocalsid".
36  # If not defined, parameter is taking from "net getlocalsid" return
37  SID="S-1-5-21-4251291460-501516116-4086100184"
38
39  # Domain name the Samba server is in charged.
40  # If not defined, parameter is taking from smb.conf configuration file
41  # Ex: sambaDomain="IDEALX-NT"
42  sambaDomain="PLANCK"
43
44  ##############################################################################
45  #
46  # LDAP Configuration
47  #
48  ##############################################################################
49
50  # Notes: to use to dual ldap servers backend for Samba, you must patch
51  # Samba with the dual-head patch from IDEALX. If not using this patch
52  # just use the same server for slaveLDAP and masterLDAP.
53  # Those two servers declarations can also be used when you have
54  # . one master LDAP server where all writing operations must be done
55  # . one slave LDAP server where all reading operations must be done
56  #   (typically a replication directory)
57
58  # Slave LDAP server
59  # Ex: slaveLDAP=127.0.0.1
60  # If not defined, parameter is set to "127.0.0.1"
61  slaveLDAP="127.0.0.1"
62
63  # Slave LDAP port
64  # If not defined, parameter is set to "389"
65  slavePort="389"
66
67  # Master LDAP server: needed for write operations
68  # Ex: masterLDAP=127.0.0.1
69  # If not defined, parameter is set to "127.0.0.1"
70  masterLDAP="127.0.0.1"
71
72  # Master LDAP port
73  # If not defined, parameter is set to "389"
74  masterPort="389"
75
76  # Use TLS for LDAP
77  # If set to 1, this option will use start_tls for connection
78  # (you should also used the port 389)
79  # If not defined, parameter is set to "1"
80  ldapTLS="0"
81
82  # How to verify the server's certificate (none, optional or require)
83  # see "man Net::LDAP" in start_tls section for more details
```

```
84  verify="require"
85
86  # CA certificate
87  # see "man Net::LDAP" in start_tls section for more details
88  cafile="/etc/opt/IDEALX/smbldap-tools/ca.pem"
89
90  # certificate to use to connect to the ldap server
91  # see "man Net::LDAP" in start_tls section for more details
92  clientcert="/etc/opt/IDEALX/smbldap-tools/smbldap-tools.pem"
93
94  # key certificate to use to connect to the ldap server
95  # see "man Net::LDAP" in start_tls section for more details
96  clientkey="/etc/opt/IDEALX/smbldap-tools/smbldap-tools.key"
97
98  # LDAP Suffix
99  # Ex: suffix=dc=IDEALX,dc=ORG
100 suffix="dc=max,dc=planck"
101
102 # Where are stored Users
103 # Ex: usersdn="ou=Users,dc=IDEALX,dc=ORG"
104 # Warning: if 'suffix' is not set here, you must set the full dn for usersdn
105 usersdn="ou=Users,${suffix}"
106
107 # Where are stored Computers
108 # Ex: computersdn="ou=Computers,dc=IDEALX,dc=ORG"
109 # Warning: if suffix not set here, you must set the full dn for computersdn
110 computersdn="ou=Computers,${suffix}"
111
112 # Where are stored Groups
113 # Ex: groupsdn="ou=Groups,dc=IDEALX,dc=ORG"
114 # Warning: if 'suffix' is not set here, you must set the full dn for groupsdn
115 groupsdn="ou=Groups,${suffix}"
116
117 # Where are stored Idmap entries (used if samba is a domain member server)
118 # Ex: groupsdn="ou=Idmap,dc=IDEALX,dc=ORG"
119 # Warning: if 'suffix' is not set here, you must set the full dn for idmapdn
120 idmapdn="ou=Idmap,${suffix}"
121
122 # Where to store next uidNumber and gidNumber available for new users groups
123 # If not defined, entries are stored in sambaDomainName object.
124 # Ex: sambaUnixIdPooldn="sambaDomainName=${sambaDomain},${suffix}"
125 # Ex: sambaUnixIdPooldn="cn=NextFreeUnixId,${suffix}"
126 sambaUnixIdPooldn="sambaDomainName=${sambaDomain},${suffix}"
127
128 # Default scope Used
129 scope="sub"
130
131 # Unix password encryption (CRYPT, MD5, SMD5, SSHA, SHA, CLEARTEXT)
132 hash_encrypt="CRYPT"
133
134 # if hash_encrypt is set to CRYPT, you may set a salt format.
135 # default is "%s", but many systems will generate MD5 hashed
136 # passwords if you use "$1$%.8s". This parameter is optional!
137 crypt_salt_format="%s"
138
```

```
139 ##############################################################################
140 #
141 # Unix Accounts Configuration
142 #
143 ##############################################################################
144
145 # Login defs
146 # Default Login Shell
147 # Ex: userLoginShell="/bin/bash"
148 userLoginShell="/bin/bash"
149
150 # Home directory
151 # Ex: userHome="/home/%U"
152 userHome="/home/%U"
153
154 # Default mode used for user homeDirectory
155 userHomeDirectoryMode="700"
156
157 # Gecos
158 userGecos="System User"
159
160 # Default User (POSIX and Samba) GID
161 defaultUserGid="513"
162
163 # Default Computer (Samba) GID
164 defaultComputerGid="515"
165
166 # Skel dir
167 skeletonDir="/etc/skel"
168
169 # Default password validation time (time in days) Comment the next line if
170 # you don't want password to be enable for defaultMaxPasswordAge days (be
171 # careful to the sambaPwdMustChange attribute's value)
172 #defaultMaxPasswordAge="45"
173
174 ##############################################################################
175 #
176 # SAMBA Configuration
177 #
178 ##############################################################################
179
180 # The UNC path to home drives location (%U username substitution)
181 # Just set it to a null string if you want to use the smb.conf 'logon home'
182 # directive and/or disable roaming profiles
183 # Ex: userSmbHome="\\PDC-SMB3\%U"
184 userSmbHome="\\PDC-SRV\%U"
185
186 # The UNC path to profiles locations (%U username substitution)
187 # Just set it to a null string if you want to use the smb.conf 'logon path'
188 # directive and/or disable roaming profiles
189 # Ex: userProfile="\\PDC-SMB3\profiles\%U"
190 userProfile="\\PDC-SRV\profiles\%U"
191
192 # The default Home Drive Letter mapping
193 # (will be automatically mapped at logon time if home directory exist)
```

```
194  # Ex: userHomeDrive="H:"
195  userHomeDrive="H:"
196
197  # The default user netlogon script name (%U username substitution)
198  # if not used, will be automatically username.cmd
199  # make sure script file is edited under dos
200  # Ex: userScript="startup.cmd" # make sure script file is edited under dos
201  userScript="logon.bat"
202
203  # Domain appended to the users "mail"-attribute
204  # when smbldap-useradd -M is used
205  # Ex: mailDomain="idealx.com"
206  mailDomain="max.planck"
207
208  ##############################################################################
209  #
210  # SMBLDAP-TOOLS Configuration (default are ok for a RedHat)
211  #
212  ##############################################################################
213
214  # Allows not to use smbpasswd (if with_smbpasswd == 0 in smbldap_conf.pm) but
215  # prefer Crypt::SmbHash library
216  with_smbpasswd="0"
217  smbpasswd="/usr/bin/smbpasswd"
218
219  # Allows not to use slappasswd (if with_slappasswd == 0 in smbldap_conf.pm)
220  # but prefer Crypt:: libraries
221  with_slappasswd="0"
222  slappasswd="/usr/sbin/slappasswd"
223
224  # comment out the following line to get rid of the default banner
225  # no_banner="1"
```

i parametro più importanti sono il SID (*Security IDentifier*), l'indirizzo del servente OpenLDAP, la radice dell'albero, l'uso di TLS, il metodo di cifratura delle parole d'ordine.

Per ottenere il SID bisogna eseguire il seguente comando sulla macchina in cui è attivo il servente Samba:

```
# net getlocalsid
```

si noti che l'uso di TLS è disattivato (riga 80), che il metodo di cifratura è ***crypt*** (riga 132), che l'unità organizzativa cui appartengono gli utenti è ***Users*** (riga 105) e non ***People*** come negli esempi precedenti; importante è anche la riga 126 in cui si stabilisce in quale DN memorizzare i contatori '**uid**' e '**gid**' da usare quando si creano nuovi utenti;

3. modificare il file '`smbldap_bind.conf`' inserendo il DN dell'amministratore e la relativa parola d'ordine;

4. sistemare i permessi dei file di configurazione:

   ```
   # chmod 0644 /etc/smbldap-tools/smbldap.conf
   ```

   ```
   # chmod 0600 /etc/smbldap-tools/smbldap_bind.conf
   ```

A questo punto occorre differenziare le operazioni da svolgere in base al fatto che sia utilizzato il vecchio metodo di configurazione basato sul file '`/etc/ldap/slapd.conf`' oppure la '**`cn=config configuration`**'.

9.4.1.1 Configurazione dell'elenco per Samba con il vecchio metodo

Nel primo caso si procede nel modo seguente:

1. copiare lo schema di elenco per Samba con il comando:

   ```
   # zcat /usr/share/doc/samba-doc/examples/LDAP/samba.schema.gz >↩
   ↪/etc/ldap/schema/samba.schema
   ```

2. aggiungere nel file '`/etc/ldap/slapd.conf`', tra le direttive globali, la riga:

   ```
   include     /etc/ldap/schema/samba.schema
   ```

3. per ottimizzare gli accessi ai dati in elenco da parte di Samba si possono inserire le righe seguenti fra le direttive della base di dati in '`/etc/ldap/slapd.conf`':

   ```
   index    uid,uidNumber,gidNumber,memberUid eq
   index    cn,mail,surname,givenname         eq,subinitial
   index    sambaSID                          eq
   index    sambaPrimaryGroupSID              eq
   index    sambaDomainName                   eq
   ```

4. per permettere agli utenti di cambiare la loro parola d'ordine NT e LM (*Lan Manager*) si può cambiare, sempre nello stesso file, la riga:

   ```
   access to attrs=userPassword,shadowLastChange
   ```

   in:

   ```
   access to attrs=userPassword,shadowLastChange,sambaNTPassword,sambaLMPassword
   ```

5. eseguire solo la prima delle tre operazioni elencate nel paragrafo 9.2.2 (quella che riguarda l'inserimento della radice dell'elenco, del tipo di archivio ecc.);

6. posizionarsi nella directory '`/etc/ldap`' e convertire il file di configurazione con il comando seguente:

   ```
   # slaptest -f slapd.conf -F slapd.d
   ```

9.4.1.2 Configurazione dell'elenco per Samba con il nuovo metodo

Per definire e usare lo schema per Samba con il nuovo metodo di configurazione occorre prima trasformare la sua definizione nel formato LDIF e poi inserirlo nell'elenco.

I passi da compiere sono:

1. copiare lo schema di elenco per Samba con il comando:

   ```
   # zcat /usr/share/doc/samba-doc/examples/LDAP/samba.schema.gz >↩
   ↪/etc/ldap/schema/samba.schema
   ```

2. creare il file ‘`schema_convert.conf`’ (il nome può essere anche diverso) contenente:

```
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
include /etc/ldap/schema/samba.schema
```

3. creare una directory temporanea di lavoro:

```
# mkdir /tmp/ldif_output
```

4. convertire i dati dello schema con il comando:

```
# slapcat -f schema_convert.conf -F /tmp/ldif_output -n0↩
↪-s "cn={12}samba,cn=schema,cn=config" > /tmp/cn=samba.ldif
```

5. nel file generato ‘`/tmp/cn\=samba.ldif`’ togliere dalle prime righe «{12}» in modo da ottenere:

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

6. e togliere le righe seguenti dal fondo del file:

```
structuralObjectClass: olcSchemaConfig
entryUUID: b53b75ca-083f-102d-9fff-2f64fd123c95
creatorsName: cn=config
createTimestamp: 20080827045234Z
entryCSN: 20080827045234.341425Z#000000#000#000000
modifiersName: cn=config
modifyTimestamp: 20080827045234Z
```

7. infine aggiungere lo schema all’elenco con il comando:

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

8. creare un nuovo file ‘`samba_indexes.ldif`’ con:

```
dn: olcDatabase={1}hdb,cn=config
changetype: modify
add: olcDbIndex
olcDbIndex: uidNumber eq
olcDbIndex: gidNumber eq
olcDbIndex: loginShell eq
olcDbIndex: memberUid eq,pres,sub
olcDbIndex: uniqueMember eq,pres
olcDbIndex: sambaSID eq
olcDbIndex: sambaPrimaryGroupSID eq
olcDbIndex: sambaGroupType eq
olcDbIndex: sambaSIDList eq
olcDbIndex: sambaDomainName eq
olcDbIndex: default sub
```

9. caricare gli indici definiti in questo file con il comando:

```
# ldapmodify -Y EXTERNAL -H ldapi:/// -f samba_indexes.ldif
```

9.4.2 Popolare l'elenco per Samba

A questo punto si fa ripartire il servente OpenLDAP e si può popolare l'elenco eseguendo il comando:

```
# smbldap-populate -a admin
```

Vengono inseriti tutti gli oggetti di base per il funzionamento di Samba con *backend* OpenLDAP e al termine viene richiesto di inserire la password per l'amministratore dell'elenco.

9.4.3 Configurazione di Samba

Vediamo ora le modifiche da apportare al file '`/etc/samba/smb.conf`' affinché il servente Samba funzioni in associazione a OpenLDAP.

Ovviamente si danno per acquisite tutte le configurazioni utili al funzionamento di Samba comprese quelle per la funzionalità di PDC illustrate nel capitolo 6.

Tutte le righe prese in esame fanno parte della sezione '**`global`**' del file di configurazione.

Prima di tutto si configura l'archivio degli utenti intervenendo sulla direttiva '**`passdb backend`**':

```
    passdb backend = ldapsam:ldap://127.0.0.1/
```

Quindi si impostano le opzioni relative all'elenco:

```
    ldap admin dn = cn=admin,dc=max,dc=planck
    ldap suffix = dc=max, dc=planck
    ldap group suffix = ou=Groups
    ldap user suffix = ou=Users
    ldap machine suffix = ou=Computers
    ldap idmap suffix = ou=Users
    ldap ssl = no
```

Altre impostazioni riguardano la possibilità di cambiare la parola d'ordine e di amministrare utenti e gruppi da MS-Windows e fanno uso di alcuni dei programmi del pacchetto '**smbldap-tools**':

```
    ldap passwd sync = Yes
    passwd program = /usr/sbin/smbldap-passwd %u
    passwd chat = *New*password* %n\n *Retype*new*password*↩
↪%n\n *all*authentication*tokens*updated*
    add user script = /usr/sbin/smbldap-useradd -m "%u"
    ldap delete dn = Yes
    delete user script = /usr/sbin/smbldap-userdel "%u"
    add machine script = /usr/sbin/smbldap-useradd -w "%u"
    add group script = /usr/sbin/smbldap-groupadd -p "%g"
    delete group script = /usr/sbin/smbldap-groupdel "%g"
    add user to group script = /usr/sbin/smbldap-groupmod -m "%u" "%g"
    delete user from group script = /usr/sbin/smbldap-groupmod -x "%u" "%g"
    set primary group script = /usr/sbin/smbldap-usermod -g "%g" "%u"
```

Dopo avere controllato con '**testparm**' che non ci siano errori si aggiunge la parola d'ordine dell'amministratore di Samba con il comando:

```
# smbpasswd -w psw_di_admin
```

e si fa ripartire il servente.

9.4.4 Inserimento degli utenti nell'elenco

Rimangono da inserire gli utenti (compresi gli utenti speciali corrispondenti alla macchine) per i quali si possono creare dei file LDIF in modo analogo a quanto mostrato in precedenza.

Se però, vogliamo partire da un archivio utenti GNU/Linux già esistente, possiamo migrarlo usando dei programmi Perl forniti con il pacchetto '**smbldap-tools**'.

Rendiamo tali programmi disponibili per l'esecuzione con i comandi:

```
# zcat /usr/share/doc/smbldap-tools/examples/migration_scripts/↩
↪smbldap-migrate-unix-accounts.gz > /usr/bin/smbldap-migrate-unix-accounts

# zcat /usr/share/doc/smbldap-tools/examples/migration_scripts/↩
↪smbldap-migrate-unix-groups.gz > /usr/bin/smbldap-migrate-unix-groups

# chmod 755 /usr/bin/smbldap-migrate-unix-*
```

Copiamo poi i file '`/etc/passwd`', '`/etc/shadow`' e '`/etc/group`' in '`/tmp/`' e togliamo dai file copiati le righe relative a utenti e gruppi di sistema; saranno questi file, così «depurati» a costituire l'input per la migrazione.

Per inserire utenti e gruppi nell'elenco eseguiamo (con il servente OpenLDAP attivo):

```
# smbldap-migrate-unix-groups -a -G /tmp/group

# smbldap-migrate-unix-accounts -a -P /tmp/passwd↩
↪-S /tmp/shadow
```

dopodiché si possono eliminare utenti e gruppi migrati dai file '`/etc/passwd`', '`/etc/shadow`' e '`/etc/group`'.

A questo punto non rimane che definire la parola d'ordine degli utenti con il comando:

```
# smbldap-passwd nome_utente
```

In verità la parola d'ordine per GNU/Linux sarebbe già definita (migrata da '`/etc/shadow`') ma manca quella per MS-Windows; il comando le ridefinisce entrambe (si può anche definire solo la parola d'ordine per GNU/Linux o solo quella per MS-Windows usando l'opzione '**`-s`**' o l'opzione'**`-u`**').

Ecco una parte della risposta al comando '**`slapcat`**' riguardante l'utente e il gruppo ***verdimarco*** dopo la migrazione e la definizione della parola d'ordine:

```
dn: uid=verdimarco,ou=Users,dc=max,dc=planck
objectClass: posixAccount
objectClass: inetOrgPerson
objectClass: shadowAccount
objectClass: sambaSamAccount
uid: verdimarco
cn: Verdi Marco
sn: Marco
uidNumber: 1002
gidNumber: 1002
gecos: Verdi Marco,,,
homeDirectory: /home/verdimarco
loginShell: /bin/bash
shadowLastChange: 13390
shadowMax: 99999
shadowWarning: 7
sambaSID: S-1-5-21-4251291460-501516116-4086100184-3004
structuralObjectClass: inetOrgPerson
entryUUID: aa0935aa-cc48-102a-880c-4b28893bdc7f
creatorsName: cn=admin,dc=max,dc=planck
createTimestamp: 20060830075535Z
sambaLMPassword: F1CDD48D94CE974325AD3B83FA6627C7
sambaNTPassword: 4B56A9FE1BF0C1D8F3BE14D8E86022E3
sambaPwdLastSet: 1156925821
userPassword:: e0NSWVBUfXJJOHY5R3FrOHdRN3c=
sambaAcctFlags: [UX         ]
entryCSN: 20060830083209Z#000000#00#000000
modifiersName: cn=admin,dc=max,dc=planck
modifyTimestamp: 20060830083209Z

dn: cn=verdimarco,ou=Groups,dc=max,dc=planck
objectClass: posixGroup
objectClass: sambaGroupMapping
cn: verdimarco
userPassword:: e2NyeXB0fXg=
gidNumber: 1002
sambaSID: S-1-5-21-4251291460-501516116-4086100184-3005
sambaGroupType: 2
structuralObjectClass: posixGroup
entryUUID: 4a780ca4-cc4f-102a-880f-4b28893bdc7f
creatorsName: cn=admin,dc=max,dc=planck
createTimestamp: 20060830084302Z
entryCSN: 20060830084302Z#000002#00#000000
modifiersName: cn=admin,dc=max,dc=planck
modifyTimestamp: 20060830084302Z
```

Per gestire gli utenti dell'elenco OpenLDAP per Samba si possono usare i comandi del pacchetto '**smbldap-tools**' come '**smbldap-groupadd**', '**smbldap-useradd**', '**smbldap-userdel**', il cui ruolo dovrebbe essere chiaro già dal nome.

Ad esempio per inserire in elenco il nuovo utente ***rossimario***, appartenente al gruppo omonimo, e definirne la parola d'ordine si eseguono i comandi:

```
# smbldap-groupadd -a rossimario

# smbldap-useradd -a -g rossimario -m rossimario

# smbldap-passwd rossimario
```

L'opzione '**-a**' nel primo comando provoca la creazione di un SID per quel gruppo; nel secondo comando invece fa si che quell'utente abbia anche un *account* Samba oltre che GNU/Linux.

L'opzione '**-g**' serve ad assegnare l'utente al gruppo indicato, mentre '**-m**' permette la creazione automatica della sua directory personale.

Se invece si devono inserire in elenco delle utenze per macchina (cosa essenziale se abbiamo come clienti delle *workstation* MS-Windows NT o XP) dobbiamo eseguire:

```
# smbldap-useradd -w nome_utenza
```

dove '**-w**' significa appunto che si tratta di una utenza di tipo macchina.

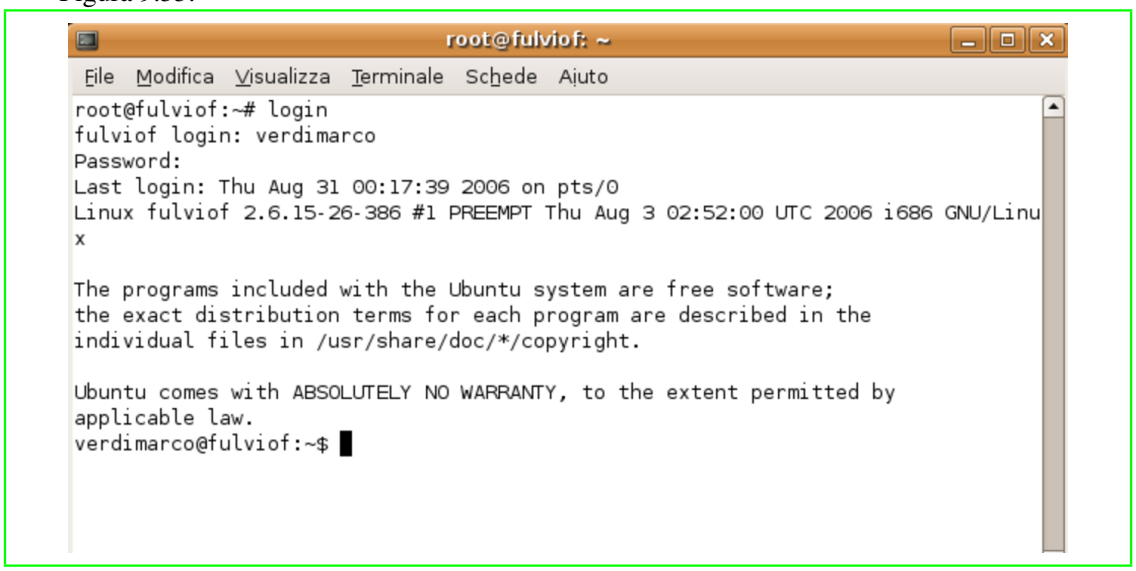
9.4.5 Uso del servente

Arrivati a questo punto si può provare a utilizzare il nostro servente Samba con *backend* OpenLDAP sia da clienti GNU/Linux configurati come illustrato nel paragrafo 9.3.1, sia da clienti MS-Windows configurati come spiegato nel capitolo 6.

A tale proposito si noti che, per unire le *workstation* MS-Windows al dominio gestito da Samba, viene richiesto di accreditarsi (durante il processo di unione effettuato sulla *workstation*) con un utente Samba con sufficienti privilegi: si può utilizzare a tale scopo l'utente ***admin***.

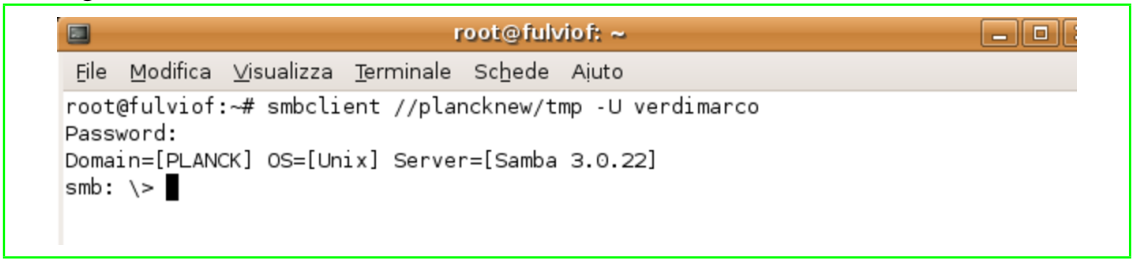
Nella figura 9.55 si vede il risultato di un tentativo di accreditamento su un cliente GNU/Linux da parte dell'utente ***verdimarco***:

Figura 9.55.



In figura 9.56 invece c'è una prova di collegamento alla risorsa condivisa ***tmp*** del servente Samba (il cui nome è ***plancknew***) effettuata con lo strumento '**`smbclient`**' e atta a verificare la validità dello stesso utente anche come *account* Samba.

Figura 9.56.



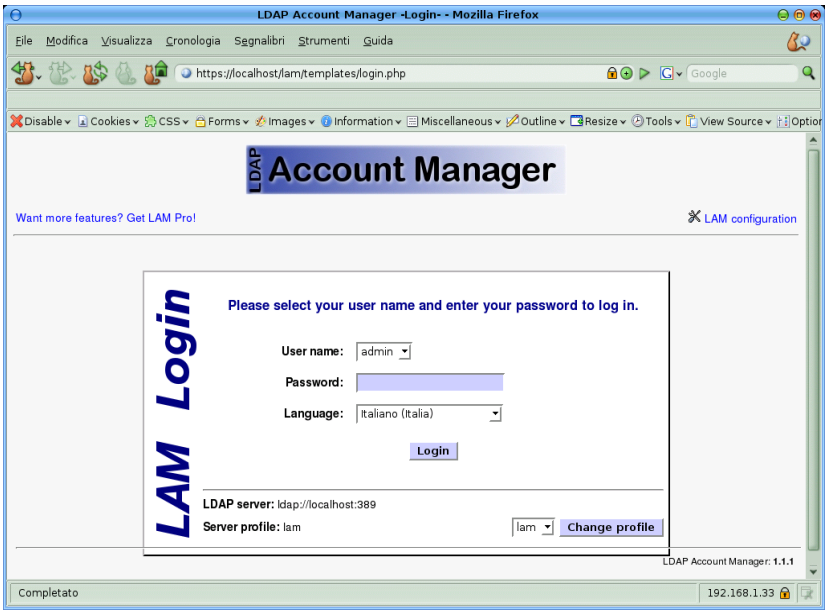
9.4.6 Interfacce grafiche per Samba-OpenLDAP

Un cliente grafico per gestire gli utenti di Samba con *backend* OpenLDAP è ldap account manager [4] (⟨*http://lam.sourceforge.net/*⟩).

Si tratta di uno strumento, scritto in PHP, che fornisce un'interfaccia Web per la gestione dei dati dell'elenco.

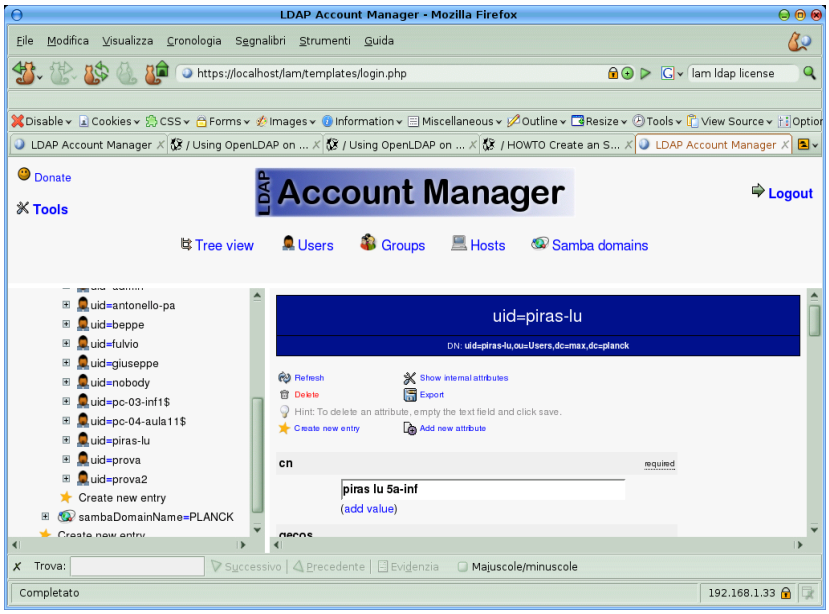
Una volta installato e configurato (per i dettagli si rimanda alla documentazione del pacchetto) si accede all'interfaccia collegandosi all'indirizzo `https://localhost/lam` e si ottiene la schermata di accreditamento (vedi figura 9.57).

Figura 9.57.



Nella figura 9.58 è invece mostrato il modulo di gestione dei dati di un utente dell'elenco.

Figura 9.58.



9.5 Usare OpenLDAP con SSL

Fino a questo momento è stato illustrato l'uso di un servente OpenLDAP con comunicazione in chiaro tra quest'ultimo e i clienti.

Questa soluzione comporta ovviamente il rischio che le parole d'ordine degli utenti cadano in mano a persone non autorizzate e tale rischio è presente anche se il servizio è utilizzato solo in un contesto di rete locale.

Per ridurre ai minimi termini tale rischio di può cifrare la comunicazione con il servente OpenLDAP usando OpenSSL.

9.5.1 Natura del pacchetto OpenSSL

OpenSSL [5] (⟨*http://www.openssl.org*⟩) permette l'utilizzo dei protocolli SSL (*Secure Socket Layer*) e TLS (*Transport Layer Security*).

SSL è stato sviluppato dalla Netscape ed è divenuto lo strumento universalmente accettato per l'autenticazione e la cifratura della comunicazione tra clienti e serventi in rete.

TLS è l'evoluzione di SSL ma la natura del protocollo non è cambiata di molto; in questo paragrafo si fa riferimento unicamente a SSL considerando il fatto che il livello di sicurezza ottenibile è del tutto analogo e che nell'uso assieme a OpenLDAP l'unica differenza visibile è nel fatto che se si usa TSL il servizio continua a rispondere sulla porta 389 mentre con SSL risponde sulla porta 636.

SSL si colloca nella pila dei protocolli TCP/IP tra il livello di trasporto e il livello applicazione e viene usato per rendere sicuri i servizi che utilizzano TCP; l'utilizzo più comune e noto anche al «grande pubblico» è quello che permette di rendere sicuro il protocollo HTTP e quindi le connessioni con i serventi Web (protocollo HTTPS).

SSL utilizza un algoritmo a chiave privata per cifrare la comunicazione tra due nodi di rete; la chiave privata viene invece trasmessa in modo sicuro grazie a un algoritmo a chiave pubblica.

9.5.2 Creazione della CA e del certificato con OpenSSL

In questo paragrafo non vengono forniti tutti i dettagli riguardanti la configurazione di OpenSSL tramite il file '`/etc/ssl/openssl.cnf`', l'uso del comando '**`openssl`**' e l'uso degli strumenti per la creazione e firma dei certificati disponibili in '`/usr/lib/ssl/misc/`'; viene solo mostrato quanto indispensabile per la creazione di certificati e chiavi per rendere sicura la comunicazione con il servente OpenLDAP.

Una volta installato il pacchetto OpenSSL con il comando:

```
# apt-get install openssl
```

occorre definire un certificato valido per il servente e quindi creare una CA (*Certification Authority*) che possa firmare tale certificato.

Questa CA non è evidentemente una di quelle globalmente riconosciute in rete e autorizzate alla diffusione dei certificati, ma solo una CA «personale».

Prima di definire il certificato possiamo impostare la durata dello stesso ad un periodo più lungo di quello impostato per difetto (1 anno); questo si fa modificando nello script '`/usr/lib/ssl/misc/CA.sh`' la seguente riga:

```
$DAYS="-days 365";
```

sostituendo il valore 365 ad esempio con 1825 (5 anni).

Creiamo poi una directory utile ad accogliere tutti i file contenenti certificati e chiavi che ci accingiamo a definire e posizioniamoci in essa (tale posizionamento rimane in essere per tutti i comandi di questo paragrafo):

```
# mkdir /etc/ssl/ca
```

```
# cd /etc/ssl/ca
```

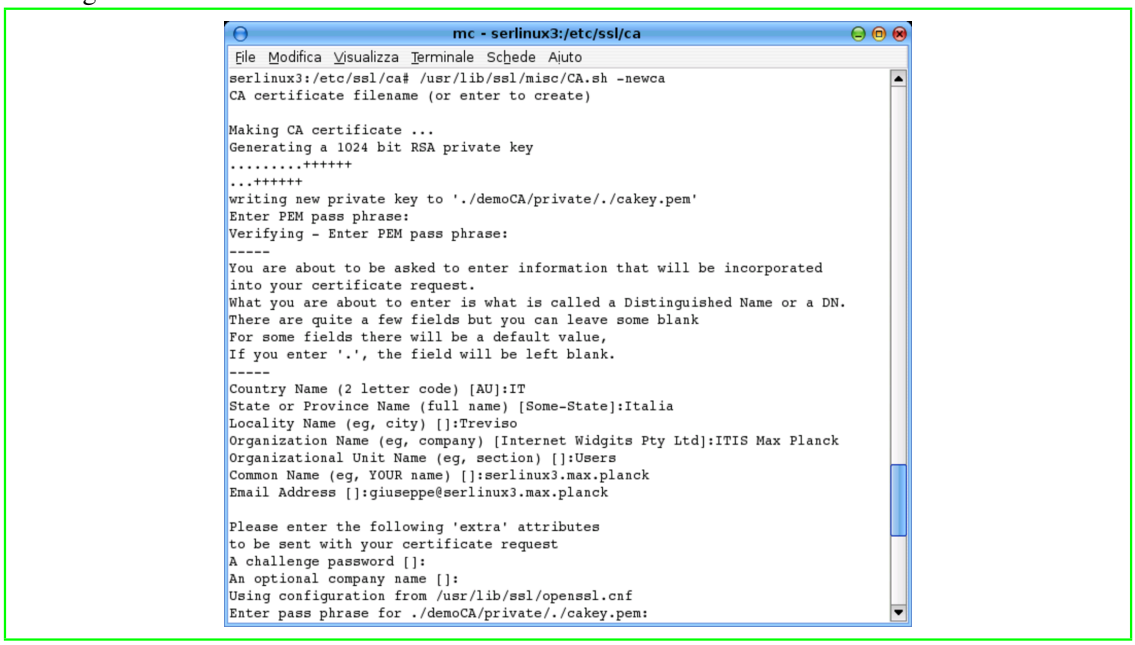
Quindi generiamo la nostra CA con il comando:

```
# /usr/lib/ssl/misc/CA.sh -newca
```

Vengono richiesti il nome di un certificato ('**`invio`**' per definirne uno nuovo); e i dati relativi alla propria organizzazione che verranno inseriti nel certificato stesso.

Nella figura 9.60 viene mostrata questa fase di lavoro con delle risposte relative ad una organizzazione di tipo scolastico (l'ITIS «Max Planck» di Treviso dove insegno).

Figura 9.60.



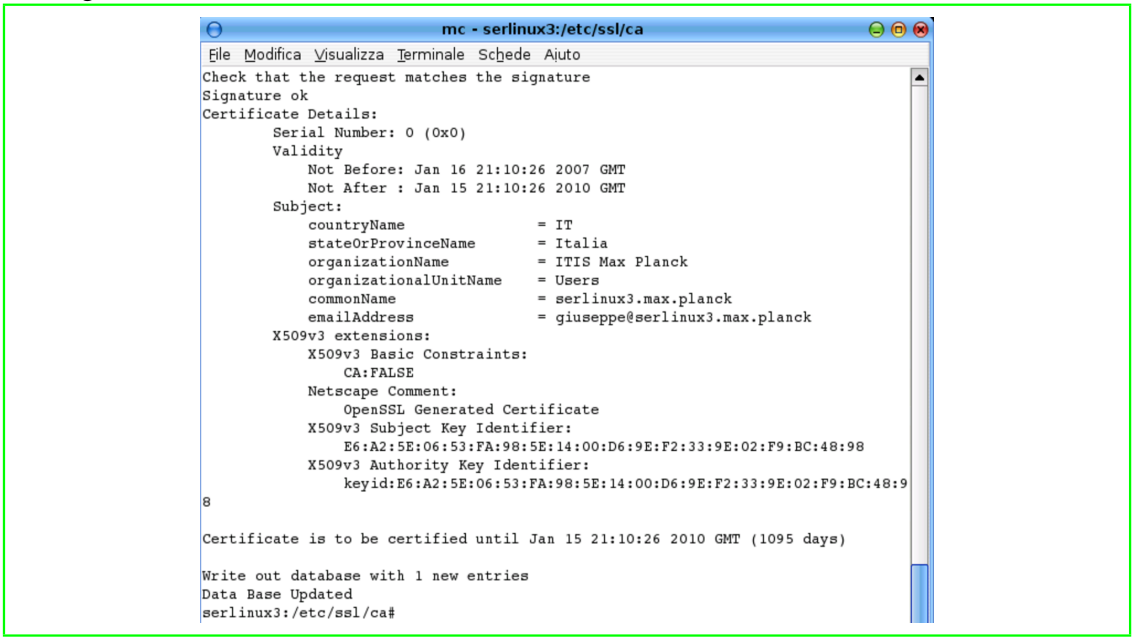
Importantissima è la risposta alla domanda riguardante il '**`common name`**': occorre indicare il FQDN *Fully Qualified Domain Name* della macchina che ospita il servente (nell'esempio ***serlinux3.max.planck***).

Sono invece opzionali e possono essere omesse le informazioni '**`challenge password`**' e '**`company name`**'.

Prestare anche particolare attenzione, all'inizio, alla definizione della '**`pass phrase`**' da associare al certificato della CA e che viene poi chiesta ogni volta che si fanno operazioni con il certificato o che se ne richiede l'uso (ad esempio al termine di questa procedura come si vede nella figura).

Nella figura 9.61 viene mostrato come si conclude il procedimento con la stampa a video dei dati del certificato; quest'ultimo con la relativa chiave pubblica viene memorizzato nel file '`/etc/ssl/ca/demoCA/cacert.pem`' mentre la chiave privata della CA nel file '`/etc/ssl/ca/demoCA/private/cakey.pem`'.

Figura 9.61.



Il passo successivo consiste nella generazione del certificato di richiesta e della chiave privata per il servente, da compiere con il comando:

```
# openssl req -new -nodes -keyout newreq.pem -out newreq.pem
```

Si ricevono le richieste di inserimento informazioni analoghe alle precedenti (figura 9.60 ma solo fino alla richiesta relativa al '**optional company name**').

Attenzione al fatto che il '**common name**' deve essere lo stesso FQDN inserito in sede di creazione del certificato della CA personale.

Il certificato di richiesta e la chiave privata vengono memorizzate nel file '`/etc/ssl/ca/newreq.pem`'.

Si può controllare la richiesta con il comando:

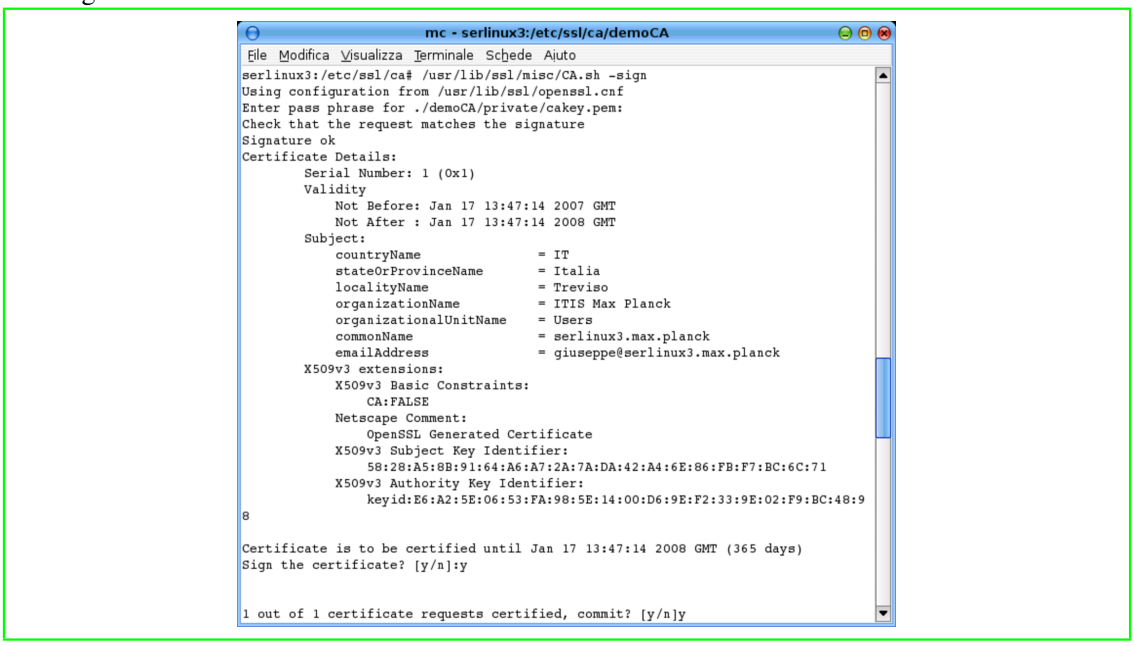
```
# openssl req -text -noout < newreq.pem
```

A questo punto serve la firma della CA sul certificato di richiesta in modo da avere un certificato valido; la si ottiene con il comando:

```
# /usr/lib/ssl/misc/CA.sh -sign
```

Nella figura 9.62 vediamo l'effetto di tale comando con la richiesta della '**pass phrase**' relativa alla CA definita in precedenza e le richieste di conferma della firma e di memorizzazione del certificato.

Figura 9.62.



Il procedimento si conclude con l'emissione a video dei dati del certificato e della chiave pubblica che vengono memorizzati nel file '`/etc/ssl/ca/newcert.pem`'.

9.5.3 Configurazione del servente OpenLDAP per l'uso con OpenSSL

Conclusa la creazione del certificato e della chiave del servente spostiamo i relativi file nella directory di configurazione di OpenLDAP cambiando loro anche il nome:

```
# cp /etc/ssl/ca/demoCA/cacert.pem /etc/ldap/cacert.pem

# cp /etc/ssl/ca/newcert.pem /etc/ldap/servercrt.pem

# cp /etc/ssl/ca/newreq.pem /etc/ldap/serverkey.pem
```

Proteggiamo poi da accessi indesiderati il file contenente la chiave privata rendendolo contemporaneamente leggibile al gruppo '**openldap**' (si tenga presente infatti che il demone '**slapd**' viene eseguito con i diritti di utente e gruppo '**openldap**'):

```
# chown root:openldap /etc/ldap/serverkey.pem

# chmod 640 /etc/ldap/serverkey.pem
```

Per configurare OpenLDAP per l'uso di SSL bisogna fare in modo che il servente sia in ascolto sia sulla porta 389 che sulla porta 636 che è quella dedicata al traffico cifrato; a tale scopo interveniamo nel file '`/etc/default/slapd`' e attiviamo la riga:

```
SLAPD_SERVICES="ldap://127.0.0.1:389/ ldaps://serlinux3.max.planck/"
```

Occorre poi intervenire sulla configurazione del servente, distinguendo, come al solito tra le vecchie versioni di OpenLDAP e quelle nuove (a partire dalla '**2.3**').

Nel primo caso si devono aggiungere le seguenti righe nel file '`/etc/ldap/slapd.conf`' tra le direttive globali:

```
TLSCACertificateFile /etc/ldap/cacert.pem
TLSCertificateFile /etc/ldap/servercrt.pem
TLSCertificateKeyFile /etc/ldap/serverkey.pem
```

Nel secondo caso si deve eseguire il comando:

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

con il quale si entra nella *console* del comando '**ldapmodify**' e si digita quanto segue:

```
dn: cn=config
add: olcTLSCACertificateFile
olcTLSCACertificateFile: /etc/ldap/cacert.pem
-
add: olcTLSCertificateFile
olcTLSCertificateFile: /etc/ldap/servercrt.pem
-
add: olcTLSCertificateKeyFile
olcTLSCertificateKeyFile: /etc/ldap/serverkey.pem
```

Al termine, dopo un ulteriore pressione del tasto [*Invio*], si ottiene il messaggio «modifying entry cn=config» e si può uscire premendo [*Ctrl c*].

Infine, in entrambi i casi presi in esame, facciamo ripartire il servizio.

9.5.4 Configurazione dei clienti GNU/Linux

Il primo file che prendiamo in considerazione sulle macchine clienti (fra le quali, come al solito, possiamo considerare anche lo stesso servente) è '`/etc/ldap/ldap.conf`' (o '`/etc/ldap.conf`') per fare in modo che i programmi lato cliente LDAP come ad esempio quelli del pacchetto '**ldap-utils**' dialoghino con il servizio sul canale cifrato:

```
BASE    dc=max,dc=planck
URI     ldaps://serlinux3.max.planck
TLS_REQCERT allow
```

Si noti l'attivazione delle interrogazioni usando il protocollo sicuro LDAPS con indicazione del FQDN della macchina servente, che deve essere lo stesso usato nella definizione dei certificati.

Inoltre è importante la direttiva '**TLS_REQCERT allow**' che fa in modo che il cliente possa richiedere il certificato della CA al servente evitando di doverne possedere una copia in locale.

Ricordarsi infine di far ripartire il demone '**nscd**'.

Per testare il buon esito delle modifiche eseguire qualche ricerca con '**ldapsearch**' sia usando il canale in chiaro:

```
# ldapsearch -x -b dc=max,dc=planck -H 'ldap://serlinux3.max.planck'
```

che quello cifrato:

```
# ldapsearch -x -b dc=max,dc=planck -H 'ldaps://serlinux3.max.planck'
```

Inoltre verificare che gli utenti gestiti dal servente siano ancora raggiungibili dal servizio NSS con il comando:

```
# getent passwd
```

9.5.5 Uso di Samba e OpenLDAP su canale cifrato

Se si vuole che anche il traffico tra i serventi Samba e OpenLDAP avvenga con l'uso di SSL occorre apportare delle modifiche ai file di configurazione di Samba e del pacchetto '**smbldap-tools**'.

Notiamo che questo è necessario solo se i due serventi sono in esecuzione su due macchine diverse altrimenti si tratta di una precauzione esagerata.
In tal caso infatti si può considerare una protezione ragionevole quella ottenibile configurando il servizio in modo che sia in ascolto sulla porta 389 (canale in chiaro, da usare per Samba) e sulla porta 636 (canale cifrato, da usare per i clienti GNU/Linux), come visto nel paragrafo 9.5.3.
Si deve infatti tenere presente che la comunicazione tra i clienti MS-Windows e il servente Samba non avviene in chiaro a meno che non si usino macchine equipaggiate con MS-Windows 95.

In '`/etc/samba/smb.conf`' bisogna aggiungere la riga:

```
ldap-ssl = on
```

e cambiare quella relativa al *backend* degli utenti in:

```
passdb backend = ldapsam:ldaps://serlinux3.max.planck/
```

In '`/etc/smbldap-tools/smbldap.conf`' bisogna attivare l'uso dei certificati variando o attivando le righe seguenti:

```
slaveLDAP="ldaps://serlinux3.max.planck"
slavePort="636"
masterLDAP="ldaps://serlinux3.max.planck"
masterPort="636"
ldapTLS="1"
verify="require"
# CA certificate
cafile="/etc/ldap/cacert.pem"
# certificate to use to connect to the ldap server
clientcert="/etc/ldap/servercrt.pem"
# key certificate to use to connect to the ldap server
clientkey="/etc/ldap/serverkey.pem"
```

Inoltre è bene fare in modo che il servente OpenLDAP sia in ascolto solo sulla porta 636 (nel contesto «ipersicuro» che stiamo definendo la 389 non serve più) e quindi variamo la riga '**SLAPD_SERVICES**' di '`/etc/default/slapd`' in modo che diventi:

```
SLAPD_SERVICES="ldaps://serlinux3.max.planck/"
```

Infine riavviamo i servizi Samba e OpenLDAP.

[1] **OpenLDAP** OpenLDAP License, 2.3

[2] **gq** GNU GPL

[3] **webmin** 3-clause BSD-style license

[4] **ldap account manager** GNU GPL

[5] **OpenSSL** OpenSSL license