

# DA DOMINIO WINDOWS A DOMINIO UNIX

## (Replicazione di Active Directory su unix per clients windows e unix)

di Giacomo Demurtas  
demurtas(at)unisi(dot)it

---

- 1: Descrizione del Progetto
  - 1.1 Samba
  - 1.2 NFS (Network File System)
  - 1.3 NIS (Network Information Service)
  - 1.4 File di Hosts o DNS
  - 1.5 Parametri di Rete
- 2: Installazione e configurazione del Server per Windows
  - 2.1 Configurazione Iniziale
  - 2.2 I pacchetti Samba
  - 2.3 Il file smb.conf
  - 2.4 Le Share Samba
  - 2.5 Il file di startup per windows e ultimi accorgimenti
- 3: Installazione e configurazione del Client Windows
  - 3.1 Modifiche al registro di sistema
  - 3.2 Il Join al nuovo Dominio
- 4: Installazione e configurazione del Server per Unix
  - 4.1 Il file di host
  - 4.2 I Pacchetti NIS
  - 4.3 Configurazione del server NIS
  - 4.4 I Pacchetti NFS
  - 4.5 Configurazione delle export NFS
  - 4.6 Come funziona l'autofs
- 5: Installazione e configurazione del Client Unix
  - 5.1 Installazione dei Pacchetti client
  - 5.2 Configurazione del client
- 6: Tweak e settaggi aggiuntivi
  - 6.1 Aggiungere, eliminare utenti e cambiare le passwords
  - 6.2 Dismettere una macchina windows
  - 6.3 Mappare i Gruppi Windows/Unix
  - 6.4 Uniformare il profilo utente windows e linux

## 1: Descrizione del Progetto

In questo articolo vedremo come installare un server basato su Ubuntu Jaunty 9.04 e Samba 3 con backend di default smbpasswd perchè funga da Primary Domain Controller di una rete Windows. Lo stesso server verrà utilizzato anche per la gestione degli utenti Unix, in modo da avere un controllo sia sugli utenti Windows che sugli utenti Linux.

### 1.1: Samba

Da Wikipedia, l'enciclopedia libera.

Samba è un progetto libero che fornisce servizi di condivisione di file e stampanti a client SMB/CIFS.

Samba è liberamente disponibile, al contrario di altre implementazioni SMB/CIFS, e permette di ottenere interoperabilità tra Linux, Unix, Mac OS X e Windows.

Samba è un software che può girare su piattaforme che non siano Microsoft Windows, per esempio, UNIX, Linux, IBM System 390, OpenVMS e altri sistemi operativi. Samba utilizza il protocollo TCP/IP utilizzando i servizi offerti sul server ospite. Quando correttamente configurato, permette di interagire con client o server Microsoft Windows come se fosse un file e print server Microsoft agendo da Primary Domain Controller (PDC) o come Backup Domain Controller, può inoltre prendere parte ad un dominio Active Directory.

Samba, quindi, è un pacchetto software che dà all'amministratore flessibilità e libertà in termini di installazione, configurazione, e scelta di sistema operativo e hardware. Grazie a questa ampia offerta, Samba è diventato molto popolare, e continua ad esserlo, ogni anno sempre di più fin dal suo rilascio nel 1992.

Il nome "samba" deriva dall'inserzione di 2 vocali nel nome del protocollo usato da Microsoft "SMB" (server message block). Originariamente si chiamava smbserver ma fu cambiato a causa dell'azienda "Syntax" che vendeva un prodotto chiamato TotalNet advanced Server e possedeva il marchio SMBserver.

Samba configura share (condivisioni) per le directory UNIX scelte (incluse le sottodirectory). Queste appaiono agli utenti di Microsoft Windows come normali cartelle di Windows accessibili via rete. Gli utenti Unix possono effettuare il mount delle directory condivise direttamente nell'albero del filesystem o, in alternativa, è possibile usare una utility, smbclient installata con samba per leggere le directory condivise mediante un'interfaccia simile a quella del programma FTP via linea di comando. Ogni directory può avere differenti privilegi di accesso sovrapposti ai normali privilegi UNIX. Per esempio: le home directory potrebbero essere accessibili in lettura/scrittura a tutti gli utenti riconosciuti dal sistema permettendo ad ogni utente di accedere ai propri file. Comunque potrebbero non avere l'accesso ai file degli altri utenti a meno che questo permesso non sia normalmente previsto. Notare che /etc/samba/netlogon, tipicamente distribuita come condivisione in sola lettura, è la directory di logon per gli script di logon dell'utente.

La configurazione di Samba si ottiene modificando un singolo file (di solito presente come /etc/smb.conf o /etc/samba/smb.conf). Samba può fornire anche script per il logon dell'utente e implementazione di policy di gruppo attraverso poledit.

Samba fu sviluppato originariamente per UNIX da Andrew Tridgell nell'Università Nazionale Australiana.

### 1.2: Network File System

Da Wikipedia, l'enciclopedia libera.

Network File System (NFS) è un protocollo sviluppato inizialmente da Sun Microsystems nel 1984 e definito dagli RFC 1094, 1813, (3010) e 3530. L'NFS è un file system che consente ai computer di utilizzare la rete per accedere ai dischi remoti come fossero dischi locali.

La versione 2 del protocollo originale utilizzava unicamente UDP e prevedeva che i server non dovessero conservare memoria degli accessi degli utenti. In passato quindi eventuali meccanismi di blocco delle risorse (Lock) dovevano essere implementati esternamente al protocollo.

La versione 3 introdusse il supporto per il TCP per il trasporto delle informazioni. Molte società avevano comunque già esteso l'NFSv2 per supportare il TCP in trasmissione. L'utilizzo del TCP consente di utilizzare l'NFS attraverso una WAN sebbene non sia molto semplice né efficiente.

La versione 4 venne influenzata dall'AFS e includeva miglioramenti nelle prestazioni, aggiungeva un supporto migliorato alla sicurezza e introduceva un protocollo

che teneva conto dello stato dei client.

NFS è spesso associato a sistemi Unix, sebbene spesso sia utilizzato anche da macchine utilizzanti i sistemi operativi Macintosh e Microsoft Windows. Il server message block (SMB) e protocolli similari sono un'implementazione simile di un file system di rete per computer Microsoft Windows. Il termine "network file system" ormai viene utilizzato come termine generico per indicare un File System in grado di gestire dispositivi di memorizzazione remoti.

Il NFS permette ad n calcolatori che compongono un sistema distribuito di condividere file, directory od un intero file system utilizzando il protocollo client-server. Un calcolatore (client) deve richiedere esplicitamente ad un altro calcolatore (server) del sistema distribuito di condividere una directory o un file, dichiarando un punto di montaggio. Una volta effettuato un montaggio un utente sul calcolatore client accede alla directory montata in modo assolutamente trasparente, ossia accede alla directory remota credendo di accedere ad una directory locale.

---

### 1.3: Network Information Service

Da Wikipedia, l'enciclopedia libera.

Network Information Service o NIS (originariamente chiamato Yellow Pages o YP) è un protocollo per servizi directory client-server per la distribuzione dei dati di configurazione di sistema, come utenti e nomi host, tra computer su una rete. Estratto da "[http://it.wikipedia.org/wiki/Network\\_Information\\_Service](http://it.wikipedia.org/wiki/Network_Information_Service)"

---

### 1.4: File di Hosts o DNS

Per realizzare questo progetto avremo anche bisogno di un servizio che ci faccia vedere nella nostra rete quali sono le macchine (server, clients stampanti ecc) presenti. Per fare questo, se usiamo indirizzi assegnati dal QIT, nulla ci vieta di usare i server DNS dell'università, se invece usiamo una nostra sottorete o se anche soltanto vogliamo restringere il campo di ricerca delle macchine disponibili, dobbiamo farci un nostro server DNS o più semplicemente condividere il file di hosts con i nostri client via NIS.

---

### 1.5: Parametri di rete utilizzati

In tutta la guida saranno utilizzati i seguenti parametri per la configurazione della rete:

- Server Linux 1
  - Nome della macchina: srvlnx1
  - Nome del dominio: GNUUNISI
  - Nome del dominio NIS: gnuunisi
  - Nome NETBIOS del dominio: srvlnx1
  - Classe IP: 192.168.1.0 / 255.255.255.0
  - IP macchina: 192.168.1.151
  - Nome utente di base: gnu
  - Password di gnu: gnu
- Client Linux 1
  - Nome della macchina: cllnmx1
  - Nome del dominio NIS: gnuunisi
  - Classe IP: 192.168.1.0 / 255.255.255.0
  - IP Server: 192.168.1.156
  - Nome utente di base: gnu
  - Password di gnu: gnu
- Client Windows 1

- Nome della macchina: cliwin1
- Nome del dominio: GNUUNISI
- Classe IP: 192.168.1.0 / 255.255.255.0
- IP macchina: 192.168.1.155
- Parametri Generali
  - Gruppo di Dominio: gnuugroup
  - gid: 3000
  - Utente di Dominio: gnuser
  - Password: gnuser
  - uid: 3001

Questi parametri vanno ovviamente adattati alle vostre esigenze.

## 2: Installazione e configurazione del Server per Windows

In questa parte vedremo come installare un server basato su Ubuntu Jaunty 9.04 e Samba 3 con backend di default smbpasswd perchè funga da Primary Domain Controller di una rete Windows. Armatevi di pazienza e state attenti ad ogni passo che faremo, un minimo errore può compromettere il funzionamento del sistema.

### 2.1 Configurazione Iniziale

Per prima cosa procurarsi Ubuntu Jaunty 9.04 versione Server. Installarla al minimo, apriamo con un editor di testo il file `/etc/network/interfaces` ed assicuriamoci che la nostra scheda di rete abbia un indirizzo ip statico.

```
...
# The primary network interface
auto eth0
iface eth0 inet static
address 192.168.1.151
netmask 255.255.255.0
gateway 192.168.1.1
...
```

Impostiamo i server DNS su `/etc/resolv.conf`:

```
nameserver 193.205.4.2
nameserver 193.205.4.11
```

Impostiamo l'hostname `/etc/hostname` giusto (`srv1nxi`) se non l'abbiamo fatto in fase di installazione:

Riaviamo il servizio di rete:

```
# sudo ifup eth0
# sudo /etc/init.d/networking restart
# sudo apt-get update
# sudo apt-get install openssh-server
```

Il pacchetto `openssh-server` servirà a controllare da remoto il nostro server.

### 2.2 I pacchetti Samba

Partiamo subito dall'installazione dei pacchetti Samba, quindi digitiamo i seguenti comandi:

```
# sudo apt-get install samba libpam-smbpass
```

Il pacchetto libpam-smbpass serve per sincronizzare le password samba con quelle UNIX.

---

### 2.3 Il file `smb.conf`

Configuriamo Samba modificando `/etc/samba/smb.conf`. Il `security mode` va settato come `user`, e `workgroup` sarà il nome del nostro dominio:

```
...
workgroup = GNUUNISI
netbios name = srv\lrx1
...
security = user
...
```

Nella sezione "Domains" aggiungere o decommentare le seguenti righe:

```
...
domain master = yes
domain logons = yes
logon path = \\%\%\%U\profile
logon drive = H:
logon home = \\%\%\%U
logon script = logon.cmd
add machine script = /usr/sbin/useradd -N -g domcompu -c "%u machine account" -d /var/lib/samba -s /bin/false -K UID_MIN=2001 -K UID_MAX=2999 %u
...
```

- `domain master`: forza il server a diventare il PDC se ci sono più domini nella rete.
- `domain logons`: fornisce il servizio di netlogon che permette a Samba di comportarsi come controller di dominio.
- `logon path`: è l'indirizzo in rete dove sarà salvato il profilo di Windows in questo caso in una cartella "profile" dentro ogni home directory. Se si preferisce è anche possibile creare un punto di condivisione samba [profiles] dove mettere tutti i profili in un'unica directory.
- `logon drive`: specifica l'unità di rete Windows dove montare le home directory.
- `logon home`: specifica il percorso delle home directory.
- `logon script`: questo è lo script che verrà eseguito localmente una volta che gli utenti si saranno autenticati. Lo script deve essere messo nella condivisione [netlogon].
- `add machine script`: questo script crea automaticamente il "Machine Trust Account" necessario alla workstation per connettersi al dominio. Lo switch `-N` non fa creare un nuovo gruppo con il nome uguale all'user

[Nota]

Se non vuoi usare i Roaming Profiles lascia commentate le opzioni "logon home" and "logon path".

---

## 2.4 Le Share Samba

Sempre nel file `/etc/samba/smb.conf`. Editare i seguenti files per abilitare le condivisioni Samba:

```
[homes]
comment = Home Directories
browseable = no
read only = no
create mask = 0700
directory mask = 0700
valid users = %S

[netlogon]
comment = Network Logon Service
path = /srv/samba/netlogon
guest ok = yes
read only = yes

[gnushare]
comment = Directory Scrivibile e leggibile da tutti
path = /gnushare
guest ok = yes
read only = no
```

Ogni volta che un client si connette viene creata una copia del `smbd` daemon, è quindi possibile fare in modo che ogni client che si connette al server abbia la propria configurazione personalizzata. Per fare questo Samba permette di inserire delle variabili nel file `smb.conf`.

Dentro al file `file smb.conf`, una variabile inizia col simbolo percentuale (%) seguita da una singola lettera maiuscola o minuscola, e può essere usata come di seguito descritto.

Per esempio:

```
[pub]
path = /home/ftp/pub/%a
```

La variabile `%a` sarà rimpiazzata dalla stringa rappresentante l'architettura del client:

- Win95 Windows for Workgroups
- Win95 Windows 95 and 98
- WinNT Windows NT
- Win2K Windows 2000 and XP
- Samba Samba
- Unknown Any other OS

There are 20 variables as shown below:

```
Client Variables
• %a Client's architecture (see above)
• %I Client's IP address
• %m Client's NetBIOS name
• %M Client's DNS name

User Variables
• %u Current Unix username
• %U Requested client username (not always used by Samba)
• %H Home directory of %u
• %G Primary group of %u
• %g Primary group of %U

Share Variables
• %S Current share's name
• %P Current share's root directory
• %p Automounter's path to the share's root directory (if different from %P)

Server Variables
• %d Current server process ID
• %h Samba server's DNS hostname
• %L Samba server's NetBIOS name
• %N Home directory server, from the automount map
• %v Samba version

Miscellaneous Variables
• %R The SMB protocol level that was negotiated
• %T The current date and time
• %$var The value of environment variable var
```

Digitiamo il seguente comando per verificare che il file di configurazione di samba sia corretto:

```
# sudo testparm
```

Ora creiamo la directory necessaria, ed il file vuoto (per ora) logon.cmd:

```
# sudo mkdir -p /srv/samba/netlogon
# sudo touch /srv/samba/netlogon/Logon.cmd
# sudo mkdir /gnushare
```

---

## 2.5 Il file di startup per windows e ultimi accorgimenti

Nel file logon.cmd è possibile mettere qualsiasi comando shell Windows per personalizzare l'accesso degli utenti. (Montaggio unità di rete, esecuzione di programmi o altri script in locale ecc. ecc.).

E' necessario creare il gruppo "domcompu" che raccoglierà le informazioni delle workstations e per poter connettere la workstation al dominio, un gruppo UNIX (es:domadmin) dovrà essere mappato con il gruppo "Windows Domain Admins".  
Cambiate pure il gruppo domcompu con qualsiasi preferite. Naturalmente deve essere chiaro che l'utente che sarà utilizzato per collegarsi al Dominio dovrà far parte di questo gruppo.

Digitate da un terminale questi comandi:

```
# sudo addgroup domcompu --gid 515
# sudo addgroup domadmin --gid 512
# sudo net groupmap add ntgroup="Domain Admins" unixgroup=domadmin rid=512 type=d
# sudo usermod -a -G domadmin gnu
```

Creare ora un gruppo ed un utente di prova:

```
# sudo addgroup gnugroup --gid 3000
# sudo useradd -g gnugroup -u 3001 -m -s /bin/bash gnuser
# sudo passwd gnuser
# sudo smbpasswd -a gnuser
```

Inserire come richiesto la password per nostro utente di test. Il comando `smbpasswd` serve solo per mettere l'utente nel dominio, poi per cambiare la password basterà il comando `passwd`.

Ora è necessario riavviare il server per utilizzare il nuovo controller di dominio:

```
# sudo reboot
```

### 3: Installazione e configurazione del Client Windows

Ora dovrebbe essere possibile far connettere il clients Windows al dominio appena creato nello stesso modo con cui lo possiamo connettere ad un dominio su Windows Server.

Attenzione: non è possibile accedere a un dominio con Windows XP Home Edition.

#### 3.1 Modifiche al registro di sistema

Windows 2000 può fare join al dominio con il metodo tradizionale, mentre in Windows XP è necessario modificare una chiave del registro con regedit (click su Start->Esegui->regedit)

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\netlogon\parameters  
"RequireSigsal"=dword:00000000
```

dopodiché si può fare join in maniera tradizionale.

#### 3.2 Il Join al nuovo Dominio

Andare su Pannello di Controllo -> Sistema -> Nome Computer -> Cambia, click su dominio e digitare gnuusr, alla richiesta di username e password digitare gnu (il nostro Domain Administrator).  
Premere OK e riavviare il PC.

## 4: Installazione e configurazione del Server per Unix

In questa parte ci occuperemo di installare il server NIS ed NFS per poter servire anche i clients linux in modo che l'utente usi la stessa password sia per loggarsi con windows che con linux.

Per fare tutto quello che sarà descritto in seguito si presuppone che sia i server che i client abbiano indirizzi IP statici, questo perchè il NIS con gli IP dinamici ha grossi problemi di sicurezza.

### 4.1 Il file di host

E' necessario aggiungere le informazioni di ogni client (nome e IP) nel file `/etc/hosts` anche l'IP del server deve esserci, non intendo il `127.0.0.1` ma l'ip disponibile ai clients. Questo ci permetterà di far funzionare il riconoscimento delle macchine della nostra rete anche se il server DNS smette di funzionare. Se preferite potete saltare questo passaggio e affidarvi solo al server DNS.... a vostro rischio e pericolo.

```
127.0.0.1    localhost
127.0.1.1    srvlnx1
192.168.1.155 cliwin1
192.168.1.156 clilnx1
...
```

### 4.2 I Pacchetti NIS

Installiamo il NIS:

```
# sudo apt-get install portmap nis
```

Sarà richiesto di mettere il nome del dominio NIS, il nostro è `gnunisi`, a questo punto il server si fermerà per qualche secondo e andrà in timeout cercando di fare un bind ad un dominio non ancora attivo.

### 4.3 Configurazione del server NIS

Editare il file `/etc/default/portmap` ed assicurarsi che la linea `OPTIONS="-i 127.0.0.1"` sia commentata.

Editare il file `/etc/default/nis` e mettere la linea `NISSERVER a NISSERVER = master`

Editare il file `/etc/yp.conf` e aggiungere le informazioni del server nel seguente modo:

```
...
domain gnutixi server srvlrx1
...
```

A chi servisse.... le informazioni del nome di dominio sono scritte nel file /etc/defaultdomain

Editare il file /var/yp/makefile e modificare o aggiungere le seguenti righe per propagare users e gruppi dal minimo id che vogliamo:

```
...
MINUID=3000
MINGID=3000
...
```

Editare ora il file /etc/ybserv.secrenets e aggiungere le linee per restringere l'accesso ai membri del dominio. Io ho preferito usare una linea per ogni host, ma è possibile usare anche le sottoreti:

```
host 192.168.1.151
host 192.168.1.155
host 192.168.1.156
```

**IMPORTANTE!!!:** Commenta la linea 0.0.0.0 . Altrimenti chiunque potrà accedere.

E' ora di costruire il nostro database:

```
# sudo /usr/lib/yp/ypinit -m
```

Seguire le istruzioni che ci verranno mostrate a video, probabilmente appariranno diversi errori, se così fosse r eseguite il comando.

Riavviare i servizi

```
# sudo /etc/init.d/portmap restart
# sudo /etc/init.d/nis restart
```

Possiamo testare il funzionamento del server lanciando il comando `ypcat passwd` o `ypcat group` e vedremo le informazioni che il server nis esporta

Ora in poi ogni volta che modifichiamo o aggiungiamo utenti gruppi oppure hosts, per aggiornare le mappe del server NIS dobbiamo lanciare il comando:

```
# sudo make -C /var/yp
```

---

#### 4.4 I Pacchetti NFS

```
# sudo apt-get install portmap nfs-kernel-server
```

---

#### 4.5 Configurazione delle export NFS

Per comodità d'uso configureremo un netgroup che è praticamente una mappa distribuita via NIS che contiene un'elenco di macchine, modifichiamo il file `/etc/netgroup` nel modo seguente:

```
gnucomputers (srv\lnx1,,) (cli\lnx1,,) (cliwin1,,)
gnuclients (cli\lnx1,,) (cliwin1,,)
```

E aggiorniamo il database:

```
# sudo make -C /var/yp
```

Il file `/etc/exports` contiene le informazioni delle directory da condividere via NFS:

```
/home          @gnuclients(rw, sync, no_subtree_check)
/srv/samba/netlogon @gnuclients(rw, sync, no_subtree_check)
/gnushare      @gnuclients(rw, sync, no_subtree_check)
```

`rw` permette di leggere e scrivere sulla condivisione (visti i permessi dell'utente naturalmente), e il `sync` forza il server a rispondere solo alle richieste dopo che ogni cambiamento sia stato fatto esattamente sul disco. Questa è l'impostazione più sicura, `async` è molto più veloce ma anche dannosa, facendo un `man exports` possiamo capire l'utilità di questi switches.

A questo punto esportiamo le condivisioni:

```
# sudo exports -ra
```

Questo deve essere fatto ogni volta che si modifica il file `/etc/exports`.

Riavviamo i servizi:

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

---

#### 4.6 Come funziona l'autofs

L'Automount e l'Autofs sono due modi per montare le directory automaticamente a richiesta. Gli Automounts sono montati solo quando ci si accede, e possono essere smontati automaticamente dopo un periodo di inattività. In questo modo gli automount NFS/Samba risparmiano banda e offrono migliori performances rispetto agli automount statici montati via fstab. Siccome dovremmo migrare le mappe di automount via nis, sul server creeremo solo i file di configurazione e il servizio lo installeremo unicamente sui clients.

Modifichiamo o creiamo il file `/etc/auto.master`, come si può ben vedere ogni mount point è associato con un'altro file di configurazione, aggiungiamo la seguente riga:

```
...
/gnushare      /etc/auto.local
/home          /etc/auto.home
...
```

Questo crea due mount points `/home` e `/gnushare` e li configura seguendo i settaggi specificati in `/etc/auto.home` e `/etc/auto.local` che dobbiamo creare e modificare come segue:

```
# /etc/auto.local
srvlnx1      srvlnx1:/gnushare

# /etc/auto.home
*           srvlnx1:/home/&
```

Questi files contengono una linea per ogni condivisione con al seguente sintassi : {mount point} [{mount options}] {location}.

Il file `auto.home` come si vede è un po' diverso dalla sintassi appena descritta, questo perchè è scomodo scrivere un file del genere:

```
# /etc/auto.home
user1      server:/home/user1
user2      server:/home/user2
user3      server:/home/user3
```

Naturalmente funziona, ma con la sintassi che abbiamo messo, l'asterisco `*` serve per piazzare sul mountpoint tutte le directory trovate nella posizione dell'e commerciale & creando tanti punti di montaggio quanti sono gli users e montando solo quelli che lo richiedono.

Editiamo ora il file `/var/yp/Makefile` e modificare o aggiungere la seguente riga per propagare le mappe di automount:

```
Aggiorniamo il database:
ALL += auto.master auto.home auto.local
...
```

```
# sudo make -C /var/yp
```

Digitare `ypcat auto.master` per vedere se abbiamo le mappe esportate e il gioco è fatto.

Un ultimo accorgimento, assicuriamoci che i servizi si avvino nell'ordine giusto, installare il pacchetto `sysvconfig`:

Lavoriamo con questo tool per far partire (i servizi con la S) nell'ordine portmap, nfs-common, nfs-kernel-server, nis:

```
# sudo apt-get install sysvconfig
```

```
# sudo sysvconfig
```

## 5: Installazione e configurazione del Client Unix

Siamo all'ultima parte, il client Linux, prendiamo una macchina fresca di installazione e con pochi semplici passaggi dovremo riuscire a vedere tutto quello che vede il client Windows, e naturalmente a loggarci con le stesse credenziali.

### 5.1 Installazione dei Pacchetti client

Un consiglio agli amministratori: se non esiste l'account root, assicuratevi che chiunque deva amministrare la macchina sia scritto ne file `/etc/sudoers` della macchina client. Un'altra buona idea è quella di avere lo stesso user anche migrato via dominio con lo stesso UID, in modo che nell'eventualità che il dominio cada sia sempre possibile amministrare la macchina localmente.

Per prima cosa procurarsi Ubuntu Jaunty 9.04. Installarla al minimo, apriamo con un editor di testo il file `/etc/network/interfaces` ed assicuriamoci che la nostra scheda di rete abbia un indirizzo ip statico.

```
...
# The primary network interface
auto eth0
iface eth0 inet static
address 192.168.1.156
netmask 255.255.255.0
gateway 192.168.1.1
...
```

Impostiamo i server DNS su `/etc/resolv.conf`:

```
nameserver 193.205.4.2
nameserver 193.205.4.11
```

Impostiamo l'hostname `/etc/hostname` giusto se non l'abbiamo fatto in fase di installazione:

```
cli1nx1
```

Riavviamo il servizio di rete:

```
# sudo ifup eth0
# sudo /etc/init.d/networking restart
```

```
# sudo apt-get install openssh-server
```

Il pacchetto openssh-server servirà a controllare da remoto il nostro server.

Aggiungi il server nel file `/etc/hosts` locale. In modo che sia sempre possibile raggiungere il server anche in caso di fallimento di NIS e/o DNS

```
...
192.168.1.151  srvlnx1
...
```

Installiamo insieme tutto quello che ci serve per il client NIS ed NFS:

```
# sudo apt-get install portmap nis nfs-common autofs
```

Sarà richiesto il nome del dominio nis, nel nostro caso `gmunisi`. Se per caso lo scrivi male o per qualsiasi motivo si vuole cambiare va modificato nel file `/etc/defaultdomain`

---

## 5.2 Configurazione del client

Come è successo per il server, anche il client andrà out of time per mancata risposta del server nis. Appena avremo disponibile il prompt modifichiamo così il file `/etc/yp.conf`

```
ypserver srvlnx1
```

Modifica il file `/etc/passwd` aggiungendo in fondo la riga `++++++`:

Modifica il file `/etc/group` aggiungendo in fondo la riga `+++`:

Modifica il file `/etc/shadow` aggiungendo in fondo la riga `++++++`:

A questo punto ci è rimasto solo il file `/etc/ncsswitch.conf` che gestisce le priorità di riconoscimento delle informazioni locali (cioè se le mappe di password hosts gruppi ecc vanno lette da locale, da remoto o da qualche altra parte):

```
passwd:      compat nis
group:       compat nis
shadow:      compat nis
hosts:       files nis mndns4_minimal [NOTFOUND=return] dns mndns4
netgroup:    nis
automount:   nis
```

Anche qua assicuriamoci che i servizi si avvino nell'ordine giusto, installare il pacchetto `sysvconfig`:

Lavoriamo con questo tool per far partire (i servizi con la S) nell'ordine portmap, nis, nfs-common, autofs:

```
# sudo apt-get install sysvconfig
```

```
# sudo sysvconfig
```

Per essere sicuri facciamo un bel riavvio del client:

```
# sudo reboot
```

## 6: Tweak e settaggi aggiuntivi

Ed ora qualche informazione di base utile a tutti gli amministratori... consiglio vivamente di GOOGLARE un po' per cercare altri settaggi ed informazioni che soddisfino le proprie esigenze di rete.

### 6.1 Aggiungere, eliminare utenti e cambiare le passwords

Per aggiungere un utente

```
# sudo useradd -g gnuigroup -K UID_MIN=3001 -m -s /bin/bash gnuuser2
# sudo passwd gnuuser2
# sudo smbpasswd -a gnuuser2
# sudo /etc/init.d/samba restart
# sudo make -C /var/yp
```

Per modificare la password di un utente

```
# sudo passwd gnuuser2
# sudo /etc/init.d/samba restart
# sudo make -C /var/yp
```

Per eliminare un utente

```
# sudo smbpasswd -x gnuuser2
# sudo userdel gnuuser2
# sudo /etc/init.d/samba restart
# sudo make -C /var/yp
```

Con questa opzione eliminiamo solo l'utente per eliminare anche la home directory aggiungere lo switch `-r` al comando `userdel`

### 6.2 Dismettere una macchina windows

Bisogna ricordare che quando facciamo fare join del client windows al dominio automaticamente viene creato l'utente "macchina" nel file di password. Questo perché c'è una chiamata di windows al server che viene intercettata da samba e tramite lo script `in smb.conf` viene aggiunto il trust della macchina. Questo però non succede quando la connettiamo dal dominio, perché la procedura non prevede nessun segnalamento al server (questo lo sanno bene chi usa i server Active Directory, nel caso di join di una macchina con lo stesso nome, dal server vano eliminate le info del computer client) quindi dobbiamo eliminare

queste info a mano nel seguente modo; dopo aver disconnesso la macchina dal dominio, facciamo prima un `cat /etc/passwd` per capire come si chiama la macchina che vogliamo eliminare e poi:

```
# sudo smbpasswd -x (nomemacchina)$
# sudo deluser (nomemacchina)$
# sudo /etc/init.d/samba restart
# sudo make -C /var/yp
```

---

### 6.3 Mappare i Gruppi Windows/Unix

Se avete la necessità di usare i powersers users e guests in windows con i relativi privilegi, leggetevi il documento di cui al link sotto

<http://www.samba.org/samba/docs/man/Samba-HOWTO-Collection/groupmapping.html>

---

### 6.4 Uniformare il profilo utente windows e linux

In questo caso abbiamo una installazione di linux in italiano, se fosse in inglese sostituite con i nomi corretti.

Eseguite le seguenti istruzioni SOLO DOPO avere fatto almeno un login Windows e uno Linux

Mappiamo il desktop:

```
# sudo su gnuser2
# cd /home/gnuser2
# mv Scrivania/* profile/Desktop/
# rm -R Scrivania
# ln -s profile/Desktop Scrivania
# exit
```

Mappiamo i documenti:

```
# sudo su gnuser2
# cd /home/gnuser2
# mv Documenti/* profile/Documenti/
# rm -R Documenti
# ln -s profile/Documenti Documenti
# exit
```

E così per tutte le cartelle che vogliamo.

## **Giacomo Demurtas**

L'autore di queste dispense rese possibili dopo l'esperienza acquisita lavorando sui sistemi Linux e Unix in genere (SUN e Silicon Graphics) presso il Centro NMR e collaborando con il Laboratorio QNL (Laboratorio di Neuroimmagini Quantitative, Università degli Studi di Siena) è Giacomo Demurtas.

System & Network Administrator del Centro NMR  
(Centro Servizi di Ateneo per la Risonanza Magnetica Nucleare)

Università degli Studi di Siena

c/o Policlinico Universitario S.Maria alle Scotte 2° Lotto, Piano 4s

tel. +39 0577 233403 fax. +39 0577 233120 <http://www.nmr.unisi.it> - [demurtas@unisi.it](mailto:demurtas@unisi.it)

e sono rilasciate sotto Common Creative License