



## ***Samba, che musica!***

Sezione di articoli tecnici

*Spesso le reti crescono in modo non uniforme, e ai computer originali si affiancano nuove entrate, nuovi modelli e, a volte, macchine con sistemi operativi differenti. Una tale situazione rischia di rovinare la vita ad un amministratore che decidesse di rendere omogenea la propria lan cercando di far condividere ad ogni computer le risorse disponibili.*

Potere accedere a risorse poste su differenti macchine si rivela non solo una soluzione efficace, ma spesso una comodità inaspettata, soprattutto, ad esempio, per quegli utenti abituati alle icone e al drag and drop di Windows che si troverebbero in grande difficoltà a editare, copiare o comunque ad operare su un file posto su una macchina Linux alla quale dovrebbero accedere via telnet o, peggio ancora, via ftp.

Per risolvere efficacemente un tale problema vi è una soluzione, tutto sommato semplice, gratuita e dal risultato garantito, che passa sotto il nome di Samba. Probabilmente in molti avranno sentito nominare questa suite di programmi, sviluppata da Andrew Tridgell e in continua evoluzione ad opera del Team Samba, ma forse non tutti hanno chiaro come funziona e cosa è realmente in grado di fare. Vediamo allora come opera questa suite seguendo un percorso che ci mostrerà come installare i programmi che la compongono, come configurarli e come utilizzarli.

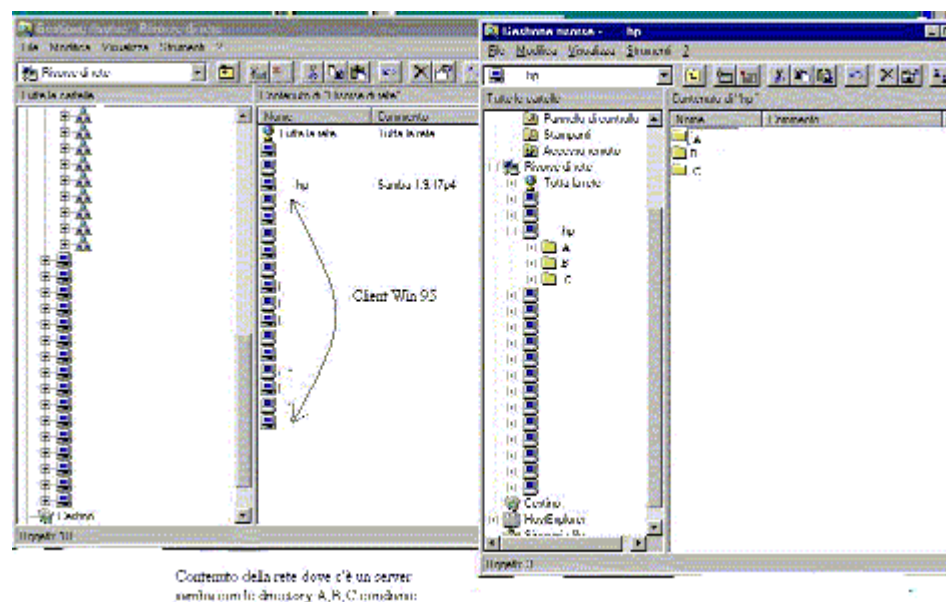
### **Protocolli e funzionamento**

Per comunicare fra di loro i computer hanno bisogno di utilizzare dei protocolli che gestiscano le operazioni di interfacciamento e consentano lo scambio reciproco di informazioni. Nel nostro caso, nella comunicazione fra differenti sistemi operativi, abbiamo a disposizione, oltre al solito tcp/ip, il protocollo netbios e l'smb. Non è questa la sede adatta per descriverne le specifiche ma è doveroso accennare quale è a grandi linee il loro funzionamento e come vengono utilizzati dalla suite Samba per potere implementare una rete mista, sia di piccole che di grosse dimensioni.

In una rete, di piccole o grandi dimensioni è indispensabile, al fine della condivisione delle risorse, il protocollo netbios che si occupa di gestire la lista dei computer in rete. In pratica, all'interno di una delle sottoreti che possono comporre la rete globale deve essere definito un master browser, ovvero una macchina che ha il compito di mantenere la lista dei computer raggiungibili da ogni client, insieme alle rispettive directory condivise. Grazie a questa postazione che contiene l'"indice" delle risorse disponibili e al protocollo smb che consente di

scambiare messaggi sullo stato delle stesse è possibile "sfogliare" attraverso le finestre di Windows la lista dei client e dei server raggiungibili dalle postazioni locale, fra i quali il server samba.

**Figura 1** Ecco come appare da Windows una rete nella quale sia presente un server Samba

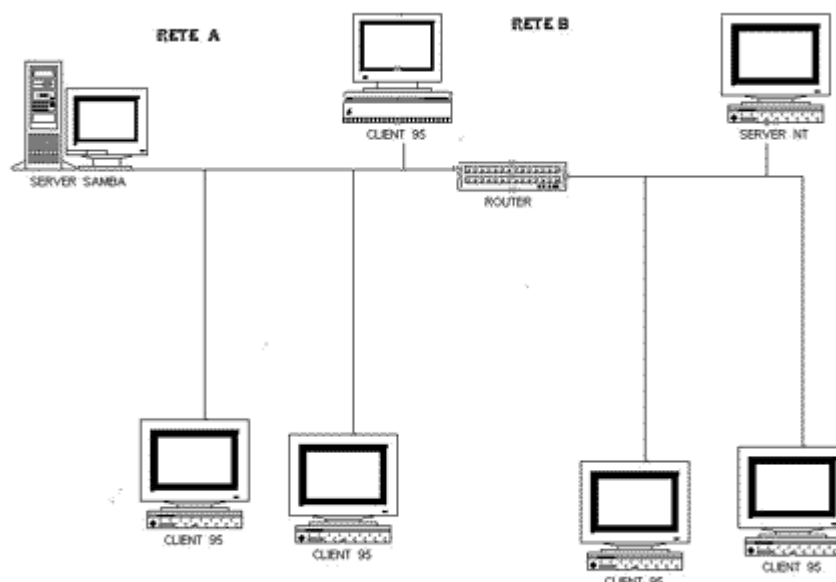


Ciò è possibile grazie anche ad una serie di comandi, definiti a livello di protocollo smb (server message blocks), i quali vengono utilizzati per passare informazioni tra i computer in rete. I messaggi sono sostanzialmente di quattro tipi:

- Messaggi di controllo della sessione, che consentono di iniziare e terminare un collegamento con una risorsa condivisa sul server.
- I messaggi di file che consentono l'accesso ai file sul server.
- Messaggi di stampanti, che sono utilizzati per inviare dati ad una coda di stampa su un server e per ottenere informazioni di stato sulla coda di stampa.
- Messaggi di messaggi, che consentono ad una applicazione di scambiare messaggi con un'altra stazione di lavoro.

Netbios, però, non è sufficiente di per se ad indirizzare dati e messaggi fra i computer di sottoreti differenti, anche se queste sono di piccole dimensioni e formate da pochi computer, come in figura 2 dove un router separa la RETE A dalla RETE B.

**Figura 2** Schema di rete con due sottoreti RETE A e RETE B, divise da un router



In questo caso interviene il protocollo tcp/ip come veicolo di trasporto per il netbios che obbliga ad assegnare almeno l'indicazione degli indirizzi ip e delle maschere di rete e quindi a pianificare i nomi dei server, o peer, e l'attribuzione degli indirizzi dei client.

Samba utilizza poi il tcp/ip come veicolo di trasporto per il netbios. Netbios soltanto funzionerebbe solo localmente in una rete di piccole dimensioni, non avremmo la possibilità di indirizzare i dati e i messaggi da un computer di una sottorete (RETE A) ad un'altra (RETE B) anche se la rete è di piccole dimensioni e formata da pochi computer come in fig. 2 dove il router separa le due reti. Il tcp/ip si occupa di indirizzare i dati utilizzando un indirizzo univoco per ogni computer e permettendo di avere reti di grosse dimensioni. Di solito, questo particolare viene dimenticato perché non sembra fare parte del protocollo netbios e di conseguenza se per sistemi Windows 95-98 basta il semplice supporto che si utilizza per le connessioni al provider, per il Dos e Windows 3.1 occorre aggiungere l'estensione tcp/ip che potrete trovare al seguente indirizzo:  
<ftp://ftp.microsoft.com/bussys/clients/>

### Come installare Samba

La prima cosa da fare è procurarsi il software e la documentazione relativa; il problema è facilmente risolvibile se avete fra le mani una qualunque distribuzione di Linux, dato che da questa potrete prelevare tutti i documenti e sorgenti da compilare di cui avrete bisogno. Se proprio non trovate ciò che vi serve, potete collegarvi al sito ufficiale di Samba <http://samba.anu.edu.au> e cercare nella directory *pub/samba*, oppure al sito <http://samba.canberra.edu.au> dove troverete diverse versioni della suite, compresa l'ultima beta, e potrete partecipare ai test; in alternativa, se preferite un collegamento veloce, potete provare il mirror italiano presente all'indirizzo <http://www.sindata.it/samba/samba.html> e raggiungibile anche attraverso un collegamento sul sito ufficiale.

Letta la documentazione, questo è il primo passo che ci sentiamo di consigliarvi, potete decidere se utilizzare i binari precompilati presenti

nell'archivio contenente la distribuzione appena scaricata, oppure compilare da soli il vostro programma. Prima di compilare i sorgenti dovreste editare il file *makefile*, togliere i commenti alle righe relative al proprio sistema operativo, nel nostro caso Linux, e lanciare il programma *make*. Fatto questo, e se non ci sono errori di compilazione basterà installare gli eseguibili e le pagine del manuale in linea. Digitando *make install* verranno installate sia le pagine del manuale che gli eseguibili, se invece si decide di fare l'installazione in due passi bisogna digitare *make installbin* e *make installman* rispettivamente per gli eseguibili e le pagine del manuale.

I passi appena descritti fanno parte della procedura standard di installazione e sono, come avete visto operazioni abbastanza semplici e alla portata dell'utente medio. Se invece volete rendere le cose ancora più semplici, recatevi sul sito della vostra distribuzione (Red Hat, Debian, Suse, Slackware, etc.), scaricate il pacchetto relativo a Samba e utilizzate gli strumenti di installazione proprietari per togliervi ogni pensiero.

Comunque abbiate installato la suite, il passo successivo, e più problematico, consiste nella configurazione del server attraverso il file *smb.conf* che troverete in

```
/usr/local/samba
```

Oppure uno degli esempi che si trovano in

```
/usr/doc/samba/examples.
```

Se proprio non riuscite a mettere le mani sul file *smb.conf*, cercate il riferimento appropriato nel *makefile* che avete utilizzato per compilare il server, oppure date un'occhiata nella directory

```
/etc
```

o, infine date il comando

```
find / -name smb.conf*
```

Dopo aver configurato il file *smb.conf*, vedremo fra poco come fare, va verificata la validità delle impostazioni e bisogna decidere con quale metodo far partire il servizio al bootstrap del computer. Per controllare la validità del file *smb.conf* è necessario lanciare un'apposita utility chiamata *testparm* che darà l'esito positivo, mostrando la configurazione corrente, oppure scriverà a video alcuni messaggi di errore.

Ecco l'output di una tipo:

```
$ testparm[Invio]
Load smb config files from /etc/smb.conf
Processing section "[public]"
Loaded services file OK.

Press enter to see a dump of your service definitions
```

Premendo [Invio] si ottiene il resto delle informazioni che riguardano la configurazione, così come è stata interpretata, completa di tutti i valori predefiniti.

Per inizializzare il servizio si può procedere in due modi o con uno script o inserendo due righe nel file `inetd.conf`. La scelta più semplice consiste nel creare uno script e collocarlo nella directory `/etc/rc.d`, in modo che all'accensione del sistema questo venga eseguito lanciando i due demoni di Samba. Ovviamente, va seguito lo standard per i nomi dei file da inserire in `/etc/rc.d`, iniziando il nome del file con una S maiuscola seguita da un numero. Ecco un esempio del file `S35samba` che contiene le istruzioni per far partire il server:

```
#!/bin/sh
# rc.samba: Start the samba server
#
if [ -x /usr/sbin/smbd -a -x /usr/sbin/nmbd ]; then
echo "Starting Samba..."
/usr/sbin/smbd -D
/usr/sbin/nmbd -D
fi
```

Una volta editato lo script, controllate che abbia i permessi di esecuzione, altrimenti dovrete definirli con il comando

```
chmod u+x S35samba
```

Un'altra cosa da controllare è che il percorso dei file riportato nello script corrisponda alla posizione reale dei file corrispondenti sul disco rigido, altrimenti non sarà possibile caricare i programmi `smbd` `nmbd`.

### Configurare `smb.conf`

Vediamo ora in dettaglio come abilitare la condivisione delle risorse di rete editando il file `smb.conf` e quali decisioni da prendere per la sicurezza delle stesse.

Come abbiamo detto precedentemente sono già presenti alcuni esempi di configurazione, che riportano i casi più comuni, che possono essere modificate a nostro piacimento per rispecchiare le nostre esigenze particolari. In ogni caso, da qualsiasi punto scegliate di partire, tenete sempre a mente il livello di sicurezza che volete implementare: Samba supporta tre opzioni per quanto riguarda la sicurezza, una delle quali va dichiarata alla voce `security`, nella sezione `global`:

```
"security= X"
```

Con X che può assumere i valori `share`, `users`, e `server`.

Se si sceglie `"security=server"` bisogna aggiungere la riga `"password server="` che deve puntare al server che si occupa della autenticazione dell'utente; in questo caso il client lavora in *user mode*. Il server può essere un'altra macchina Linux, oppure un computer dotato di sistema operativo diverso, per esempio Windows NT.

L'impostazione `security=user` è la più semplice a livello concettuale: il client dopo aver aperto una sessione manda un comando di *session setup* che contiene la combinazione `username/password` che il server può accettare o rifiutare. Se il server consente l'autenticazione dell'utente, il client può accedere a tutte le directory in share senza la richiesta di un'ulteriore password.

Con *security=share*, la modalità di default, il client si deve autenticare per ogni risorsa condivisa cui vuole accedere, inviando una password, che è diversa per ogni risorsa e indipendente dall'utente.

Volendo utilizzare il protocollo netbios e Samba per condividere dischi e stampanti senza alcun controllo sugli utenti definiamo un utente fittizio che verrà utilizzato come riferimento quando l'accesso avviene per servizi pubblici, o anonimi. Per questo, conviene aggiungere manualmente al file */etc/passwd* una riga simile a quella seguente.

```
guestpc::499:100::/dev/null:/dev/null
```

Come si può vedere, si tratta di un utente senza password, senza directory home e senza shell. Il nome *guestpc* è scelto in base al valore predefinito da Samba per questo scopo. Il numero usato come UID va scelto in modo che non coincida con altri contenuti all'interno del file */etc/passwd* per quanto riguarda la scelta del GID (il numero del gruppo), questo dipende dalle particolari strategie adottate nella gestione degli utenti. Per esempio, potrebbe essere abbinato a un gruppo omonimo *guestpc*, o simile

Un semplice *smb.conf* tratto da quello di default distribuito con la Slackware è riportato qui di seguito; ogni riga che inizia con il punto e virgola è un commento.

```
[global]
allow hosts = 192.168.1.0/255.255.255.0
;definisce la rete che può accedere ai servizi offerti da Samba
workgroup = UFFICIO
;Definisce il gruppo di lavoro al quale appartiene il server
guest account = pcquest
;Abbiamo così abilitato l'utente guest che abbiamo creato prima nel
file /etc/passwd
; Le righe che seguono servono per specificare il tipo di comando utilizzato
per la stampa e il file di spooler usato per la stampante condivisa
printing = bsd
printcap name = /etc/printcap
load printers = yes
; La riga che segue serve per avere un logfile per ogni client. Basta
commentarlo
; per avere un unico log globale
log file = /var/log/log.%m
; Togliendo il punto e virgola alle due righe sottostanti avremo una directory
pubblica con modalità di accesso share.
; lock directory = /var/locks
; share modes = yes
;Con la riga sottostante definiamo il livello di sicurezza. Al posto di
security=share si possono mettere le altre due opzioni users e server.
security=share

[public]
;definiamo un servizio denominato Public, serve per l'accesso alla
directory /home/samba
comment = directory pubblica
path = /home/samba
public = yes
;public = {yes|no} permette di definire se la directory è pubblica
writable = yes
; writable = {yes|no} definisce se gli utenti possono accedere anche in
scrittura
printable = no
; printable = {yes|no} decidiamo se è un servizio di stampa
browseable = yes

;browseable = {yes|no} permette di definire se possiamo sfogliare la directory

[homes]
comment = Home Directories
browseable = no
;browseable = {yes|no} permette di definire se possiamo sfogliare la directory
read only = no
create mode = 0750
```

```

; Maschera dei permessi che abilita il proprietario al controllo completo del
gruppo alla lettura e scrittura.

[printers]
; Definizione della sezione dedicata alle stampanti, 0700 è relativo ai
permessi sul file di spooler
comment = All Printers
browseable = no
; browseable = {yes|no} permette di definire se possiamo sfogliare la directory
printable = yes
; printable = {yes|no} decidiamo se è un servizio di stampa
public = no
; public = {yes|no} permette di definire se la directory è pubblica
writable = no
; writable = {yes|no} definisce se gli utenti possono accedere anche in
scrittura
create mode = 0700

[tmp]
; Creazione della directory condivisa per i file temporanei, togliere i
commenti per renderla attiva.
; comment = Temporary file space
; path = /tmp
; read only = yes
; public = yes
; La rendiamo pubblica

[fredsprn]
; Creazione dello spool di stampa privato utilizzabile solo dall'utente fred.
Lo spool viene messo nella home directory di Fred
; comment = Fred's Printer
; valid users = fred
; path = /homes/fred
; printer = freds_printer
; public = no
; La directory è privata, ci può stampare solo l'utente fred
; writable = no
; writable = {yes|no} definisce se gli utenti possono accedere anche in
scrittura
; printable = yes
; printable = {yes|no} decidiamo se è un servizio di stampa

```

## I programmi della suite

Gli eseguibili che si ottengono una volta compilati i sorgenti non sono solo relativi ai due demoni visti sopra e al programma di test del smb.conf. La suite presenta alcuni strumenti che ci aiutano nella gestione del nostro server e degli utenti che possono essere abilitati all'uso dei servizi messi a disposizione dalla suite Samba. Di solito, questi programmi si trovano all'interno della directory

*/usr/sbin*

Vediamo brevemente come utilizzare al meglio alcuni di questi strumenti, per gestire in maniera efficace il nostro server.

### Smbclient:

```

smbclient <server-e-servizio> [<password>] [<opzioni>]
smbclient -L <computer-host>

```

Smbclient è il programma attraverso cui è possibile connettersi a un computer che offre servizi netbios attraverso il protocollo tcp/ip, indipendentemente dal software utilizzato per il server, fosse MS-Windows 95, NT o anche computer Linux. Questa utility ci consente anche di controllare che il servizio sia effettivamente funzionante; ipotizziamo che il server stia funzionando su una macchina chiamata *hp*, possiamo verificare l'effettivo funzionamento di Samba dando il



**comando:**

```
$ smbclient -L hp [Invio]
```

In risposta si dovrebbe ottenere il risultato seguente.

```
Added interface ip=192.168.1.1 bcast=192.168.1.255 nmask=255.255.255.0
Server time is Wed Apr 9 10:47:46 1997
Timezone is UTC+1.0
Domain=[UFFICIO] OS=[Unix] Server=[Samba 1.9.16p11]
Server=[tmauro] User=[mauro] Workgroup=[UFFICIO] Domain=[UFFICIO]
Sharename Type Comment
-----
IPC$ IPC IPC Service (Samba 1.9.16p11)
lp Printer stampante pubblica
public Disk directory pubblica
This machine has a browse list:
Server Comment
-----
tmauro Samba 1.9.16p11
This machine has a workgroup list:
Workgroup Master
-----
UFFICIO mauro
```

**Smbstatus**

```
smbstatus [<opzioni>]
```

Se volete conoscere quali client sono connesse ad un dato servizio, **smbstatus** è l'ideale. Lanciatelo senza alcun parametro e date un'occhiata alla tabella che ne risulta.

**Smbmount**

```
smbmount <servizio> <mount-point> [<opzioni>]
```

**Smbmount** permette di eseguire il **mount** (ovvero collegare una risorsa remota ad una directory del proprio computer, come se fosse in locale) di una directory offerta in condivisione da un servizio netbios. Il nome del servizio viene indicato in maniera più confacente allo standard Unix, utilizzando barre inclinate normali, e non inverse come richiede normalmente netbios. Il servizio ha la seguente sintassi.

```
//<server>/<nome>[/<directory>]
```

In questo modo si indica il server e il nome del servizio, come al solito, a parte l'uso delle barre normali. **Smbmount** può eventualmente essere utilizzato anche da utenti normali (diversi da root) ma in tal caso, deve essere attivato il bit SUID (suid root), ovvero deve avere i permessi 4755 e appartenere a root.

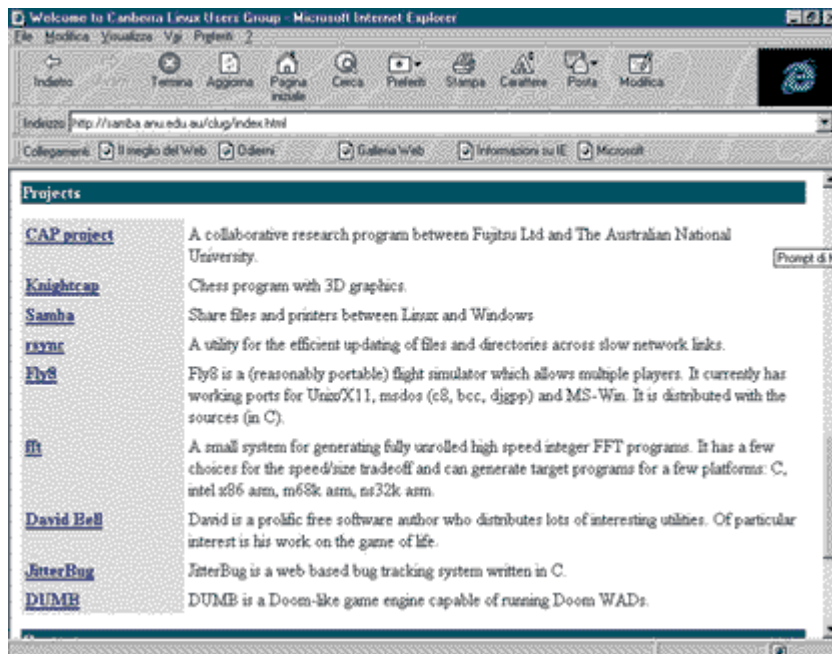
**Se qualcosa non funziona**

Se qualcosa non funziona, e se abbiamo già letto la documentazione senza trovare alcuna soluzione, possiamo provare a eseguire i test elencati nel documento *diagnosis.txt*. Se proprio non riuscite ad individuare il problema e a risolverlo, provate ad iscrivervi alla mailing list che trovate sul sito ufficiale di Samba.

**Figura 4** Se avete qualche problema, rivolgetevi alla mailing list



dedicata a Samba



Una volta risolti ogni problema, incominciate a utilizzare tutte le risorse della vostra rete senza preoccuparvi se queste risiedono su una macchina Windows o Linux: l'importante è averle, ci penserà Samba a mettervele a disposizione.

## RIFERIMENTI

Documentazione ufficiale tratta dal sito Samba  
 Appunti su Linux tratti dal sito [www.pluto.linux.it](http://www.pluto.linux.it)

E-mail  
[info@it-net99.com](mailto:info@it-net99.com)

Tutti i diritti  
 riservati da Net99

Ultimo  
 aggiornamento  
 15 ottobre 1999