

# SAMBA - PLAYING AS A PDC

Samba is a dance, but it is also something else. It is probably one of the leading reasons why linux is able to be brought into a windows network, you see samba is a piece of software which makes a linux server look like a windows one. It allows a linux server to access windows shares, and also make available windows shares, it can even function as a windows print server. But in this paper we will be going through making it even better, you see - samba can act as a windows primary domain controller (PDC)! Thats right, using a linux box you can give your windows clients a domain that they can log onto, complete with mapped drives and logon scripts. Now thats linux in action, lets take a look..

## What do you Need?

Firstly a samba server, (I am using version 2.2. Yes, I know its old, but I'm making a point) a couple of windows clients you want to add to a domain and lastly, some money you do not want to give to a monopoly. Samba has one main configuration file, *smb.conf*, this is normally found at */etc/samba*, but may be elsewhere depending on how you installed it. This configuration file is split up into the global options section and the various shares you may want to setup.

## Global

The globals section of your *smb.conf* file dictates what the baseline settings are for your server, each specific share can have different settings specified, but unless they do they use the settings as laid out in the *globals* section. Lets take a look at the *global* settings needed for a PDC..

```
1:[global]
2:    workgroup = <domain-name>
3:    netbios name = <name>
4:    server string = <name>
5:    time server = yes
6:    domain logons = yes
7:    domain master = yes
8:    domain admin group = root
9:    logon path = \\slyph\profiles\%U
10:   logon home = \\%L\%U
11:   logon script = logon.bat
12:   os level = 255
13:   local master = yes
14:   preferred master = yes
15:   wins support = yes
16:   dns proxy = yes
17:   security = user
18:   encrypt passwords = Yes
19:   smb passwd file = /etc/samba/smbpasswd
20:   kernel oplocks = no
21:   hide dot files = yes
22:   create mask = 0700
23:   directory mask = 0700
24:   log file = /var/log/samba/%m.log
25:   max log size = 10
26:   socket options = TCP_NODELAY SO_RCVBUF=8192 SO_SNDBUF=8192
```

2	The name you want to use for your domain
3,4	The name for your machine and any description you want to give it
5	If you want windows clients to be able to synchronize time with your server
6,7	Needed for the PDC functionality
8	The group which has all the domain administrator usernames
9	Where your users roaming profiles are kept. Not used if you do not use profiles
10	Each users home directory

11	This is logon script executed upon logon, if you want to customize it per user, use %U.bat
12, 13, 14	These are required to make the server the domain and browser master in your network
15	This makes your server a WINS master server
16	This helps in network browsing by clients
17	This means that your share security will be dictated by the username provided
18, 19	This needed to for security and to state where your smbpasswd file is
20- 26	General Samba options I have found useful

Believe it or not, thats it for the *global* section. There are many more options which can be useful for all sorts of things from multi-homed servers to general security to printer queues, remember to read your *smb.conf* manpage (*man smb.conf*) for all the options and what they mean. The configuration I am giving is to accomplish one purpose, the functionality of a PDC, nothing else - even though Samba is capable of much more. But now that the *global* section is done lets move onto the normal shares which are needed.

### Setting up the Homes

You should give each user a home directory which they can access from their windows machines, this generally makes everyone very happy and saves the system administrator from having to manually create a new share for each person's use. Lets take a look...

```
1:[homes]
2:  comment = Home Directories
3:  path = %H
4:  valid users = %S root
5:  force user = %S
6:  browseable = no
7:  writable = yes
8:  create mode = 0600
9:  directory mode = 0700
```

1	The share name, very important as this is how it will be known to be accessed
2	Just a comment for the share
3	The folder on the server the share points to. %H is the home directory associated with the username by the server
4	This lists who can access the server, in this case %S is the user and <i>root</i> is the <i>admin</i> group
5	Forces any access to be viewed as from the stated user
6	Not normally browseable by windows clients
7	Self-explanatory, this share can be written to
8,9	Specifies the mask used when creating new files or directories

### Setting up the System Shares

Windows clients need certain shares available in order to perform certain functions. Since we want to use logon scripts and have profiles we will need to setup shares for these features. Take a look at..

```
1:[netlogon]
2:  comment = Network Logon Service
3:  path = /var/local/netlogon
4:  admin users = root
5:  guest ok = Yes
6:  browseable = No
7:  writable = No
8:  share modes = no
```

```

9:
10:[profiles]
11:  path = /var/local/profiles
12:  browseable = no
13:  writable = yes
14:  guest ok = yes

```

1,10	The share names. These are particularly important as they dictate the function
3,11	The physical directory on the server the shares access. Make sure they exist before use with proper user rights
4	Obviously you only want the administrators to have write access here
5,14	This is to allow anyone access to the shares
6,12	This means that the shares are not normally browseable

There you go, all of the above configuration in your *smb.conf* will give you the functionality of a windows domain controller from your linux server. Fairly nifty for a couple of lines of configuration is it not? But of what use is the domain if we have no users?

### Users

In order to gain access to the domain, a user needs an account on the actual server, with a corresponding account in the samba password database. Here is a script I call *add-user* which I use add users in a single step (of course you will need to modify the *group id* and *home* variables for your setup)..

```

#
# The first value passed is the user name
# The second value passed is the password
# Example: ./add-user john longpassword
#
UADD=/usr/sbin/useradd
SADD=/usr/bin/smbpasswd
PWD=/usr/bin/passwd
GID=5001
HOME=/home

$UADD -g $GID -d $HOME/$1 -m $1
echo $2 | $PWD $1 --stdin
$SADD -a $1 $2

```

### Computers

We also need to cater for computer accounts. These accounts are used by the server to identify each client connecting. You will need both a user and a computer account in order to add a user and his machine to the domain. I use a script I call *add-pc* (very original I know) to simplify matters for myself (as always change where needed)..

```

#
# The first value passed is the pc name
# The dollar sign is very important
# Example: ./add-pc bobpc$
#
UADD=/usr/sbin/useradd
SADD=/usr/bin/smbpasswd
GID=6001
HOME=/dev/null
SHLL=/dev/null

$UADD -g $GID -d $HOME -s $SHLL $1
NAME=`echo $1 | cut -f 1 -d "$"`
$SADD -am $1

```

### A word on Windows XP

Remember two things when dealing with windows XP, firstly that XP home cannot join a domain only XP

professional can. And when joining a windows XP client first do the following..

1. Open the Local Security Policy editor (Start -> All Programs -> Administrative Tools -> Local Security Policy).
2. Locate the entry "Domain member: Digitally encrypt or sign secure channel (always)" = *Disable*
3. Locate the entry "Domain member: Disable machine account password changes" = *Disable*
4. Locate the entry "Domain member: Require strong (Windows 2000 or later) session key" = *Disable*

### *Final Words*

Simple ain't it? There in a nutshell is the configuration needed to run a linux based windows domain and two simple scripts to help add the relevant accounts needed. Yes it is simple, and yes it can do more. Thats the beauty of this solution. As always have fun and learn.