

Work with Windows users, using the Samba and the Samba Web Administration Tool.

SAMBA: SHARE WINDOWS FILES

Set up file sharing and co-exist in harmony with Windows users.

JOHN LANE

Even the most die-hard Linux fan will at some point find themselves on a network alongside users of other operating systems and will want to share files with them.

Samba is an open-source implementation of the file and print sharing protocol that Windows computers use. It was originally part of the networking suite that Microsoft implemented before they adopted TCP/IP, the networking standard that we all use today, and this legacy brings a certain quiriness to the interaction between Linux and Windows, one of which is having to deal with two name resolution services.

There are two ways that you can exchange files with a Windows system. You can, as a client, connect to another resource on the network to access files or you can set up a server to allow others to connect to you. You'll hear network-accessible filesystems being called shares and, in the Linux world, Samba shares.

Connecting to one of these as a client is very easy these days, because the drivers that you need are now part of the Linux kernel, but you may still need to install the command-line tools:

```
$ apt-get install cifs-utils
```

CIFS is the Common Internet File System, and is what Microsoft calls Samba. It was originally called Server Message Block, or SMB, which led to the Linux implementation being called Samba.

So, if all you want to do is connect to a Windows server to read and write files, it's a simple **mount**:

```
$ mount -t cifs -o username=myuser,password=mypass //myserver/myshare /mnt
```

LV PRO TIP

Apple's OS X uses the same Samba as Linux and can therefore interoperate in the same way.

The 'guest' user

When you connect to a Samba share, you do so as a specific user that, unless you specify otherwise, will be the same as your local username. The server can be configured, like our example is, to provide a guest user and to map unrecognised users to it. This allows access to permitted shares without authenticating. Shares are accessible to guests when their configuration includes: **guest ok = yes**

A quirk of the protocol requires recognised users to authenticate even when accessing shares that are accessible to guests without doing so. You can get around this by mounting with the **guest** option:

```
$ mount -t cifs -o guest //myserver/public_share /mnt
```

Our example configuration sets the ownership of files written by guests to the **nobody** user and **nogroup** group.

You'll need to be logged in as root to use **mount** like that, or you can add an entry to **/etc/fstab** to mount automatically upon boot:

```
//myserver/myshare /mnt cifs username=myuser,password=mypass,users 0 0
```

We use the **username** and **password** options to specify the credentials needed to connect to the remote share. You can omit these if you're connecting to a publicly-accessible guest share. The **users** option allows members of the **users** group to mount and unmount the share without needing root privileges.

smbclient

While you may prefer to mount shares that you frequently use, there is another way to access them that may suit for occasional use or in situations where you aren't permitted to mount. This is the **smbclient** tool, and you'll need to install it from your repository:

```
$ apt-get install smbclient
```

It works a bit like an FTP client; you connect to a host and then use **put** and **get** to send and receive files. You can give the **help** command to see the list of commands available. Here's an example session

```
$ smbclient //myhost/public
```

```
Enter john's password:
```

```
Domain=[WORKGROUP] OS=[Unix] Server=[Samba 3.6.4]
```

```
smb: \> ls
```

```
testfile N 6 Wed May 15 19:32:07 2013
```

```
smb: \> get testfile
```

```
getting file \testfile of size 6 as testfile (0.1 KiloBytes/sec)
(average 0.1 KiloBytes/sec)
```

```
smb: \> quit
```

With these methods you can read and write files shared by others, but to share yourself, you need a server – and that's where Samba comes in. There are two versions in popular use: the 3.6 series and the newer 4.x series. The major difference is that version 4 can work as an Active Directory Domain Controller, but that's overkill if all you want to do is share some files. Either version is fine for that purpose and one of them will be easily installable from your distribution's repositories:

```
$ sudo apt-get install samba
```

You configure Samba by editing its configuration file, usually `/etc/samba/smb.conf`. It is formatted similarly to the `.ini` found on Windows systems, so any text editor will do. Here is an example that provides a public share:

```
global
server string = Samba Server Version %v
# Treat unknown users as a guest (where permitted)
security = user
map to guest = Bad User

# For Windows network browsing
workgroup = LVSAMBA
netbios name = MYSERVER
name resolve order = wins bcast

tempfiles
path = /tmp
read only = No
browsable = Yes
guest ok = Yes
force user = nobody
force group = nogroup
create mask = 0755
directory mask = 0755

homes
comment = %U home directory
read only = No
browsable = No
```

Inside the Samba config file

The **global** section is for system-wide settings. Its **server string** is a description that is displayed to clients browsing the network for shares. The **“security = user”** and **“map to guest”** settings cause any unknown users to be treated as a guest. Finally, it configures the NetBIOS Workgroup. This is where the Samba server should appear in the Windows network browser (My Network Places) on Windows clients.

The **tempfiles** section describes a share called **tempfiles**, which gives access to the local `/tmp` directory. The attributes we’ve used in the example are self-explanatory; they are a few of the many available and are documented at <http://bit.ly/smbconf>. You create sections like this for each local directory that you want to share.

The **homes** section in our configuration is special because it shares users’ home directories when they authenticate using their username and password. For a user to be recognised by Samba, it needs to be created with **smbpasswd**:

```
$ smbpasswd -a myuser
Note that this sets up a separate password to that stored used by passwd. You can then use home directories
$ mount -t cifs -o username=myuser,password=mypass //myserver/myuser /mnt
```

It’s a good idea to test your configuration for errors

What is NetBIOS ?

When Microsoft implemented Windows, it used a networking API called NetBIOS (Network basic Input/Output System) that ran over various protocols, but TCP/IP wasn’t used until Windows 95 and, with Windows 2000, Active Directory began to lessen the requirement for NetBIOS, although the My Network Places browser still uses it and it allows older versions of Windows to co-exist on the same network.

NetBIOS includes several parallels to TCP/IP networking, such as the Windows Internetworking Name Server (WINS) that provides name resolution services to NetBIOS clients in a similar way to DNS. If you want to be able to resolve NetBIOS names when mounting shares, you’ll need to install **winbind**

```
$ sudo apt-get install winbind
```

and configure your systems `/etc/nsswitch.conf` to use it by adding **wins** to its **hosts** entry:

```
hosts: files wins dns
Samba implements the SMB/CIFS protocol over TCP/IP, either with (on port 139) or without NetBIOS (port 445). The nmbd daemon provides the NetBIOS services including WINS server.
```

The Samba suite gained full Active Directory compatibility in version 4, including the ability to be a domain controller, but it is unnecessary for simple file- and print-sharing.

Samba allows NetBIOS to be disabled, but doing so is only practical if Active Directory is implemented instead. Our examples keep NetBIOS, because this configuration is more likely to suit home or other small networks.

using Samba’s **testparm** command:

```
$ testparm
Load smb config files from /etc/samba/smb.conf
rlimit_max: increasing rlimit_max (1024) to minimum Windows limit (16384)
Processing section “[tempfiles]”
Loaded services file OK.
Server role: ROLE_STANDALONE
```

Samba runs two daemons, called **smbd** and **nmbd**. The former provides the sharing services and the latter provides the NetBIOS name services necessary for your Samba server to appear in My Network Places. Start the Samba daemons with:

```
$ service smbd start
$ service nmbd start
```

You should then be able to browse for the new share from a Windows machine (remember to use backslashes: `\\myserver\tmp`). Or, from a Linux (or other Unix-like) machine, you can use **findsmb** to list servers on the network and **smbclient** to view their shares.

```
$ findsmb
IP ADDR NETBIOS NAME WORKGROUP/OS/VERSION
-----
10.0.2.6 MYSERVER +[LVSAMBA] [Unix] [Samba 3.6.9]
$ smbclient -L MYSERVER
Domain=[LVSAMBA] OS=[Unix] Server=[Samba 3.6.9]
Sharename Type Comment
--- --
IPC$ IPC IPC Service (Samba Server
Version 3.6.9)
tempfiles Disk
```

We’ve covered what is necessary to access shares and provide your own. But Samba enables you to do much more, including auto-configuring home directories and sharing printers. With version 4 you can participate fully in an Active Directory network, and this is something that we will cover in a tutorial in the near future.

LV PRO TIP
You can ask Samba to reload its configuration without restarting. Use **smbcontrol all reload-config**.

THE GUI WAY TO SAMBA

Can configure and use Samba without the command prompt.

JOHN LANE

IV PRO TIP

if you log in to *Swat* as root, its status page will give you buttons to start and stop the Samba daemons.

Swat allows limited access to users without write privileges. They can view the server status and its configuration, browse documentation and change Samba passwords.

If you prefer to use a graphical configuration tool instead of manually editing files, there are various tools available that enable you to administer Samba and access remote shares without opening up a terminal window or text editor.

The first of these that we will look at is called *Swat*, or the *Samba Web Administration Tool*. It's part of the Samba suite but your distribution may package it separately from the Samba server suite. To install it on Ubuntu:

```
$ apt-get install swat
```

Before using *Swat*, bear in mind that it will rewrite Samba's configuration file `/etc/samba/smb.conf`. So, if you have carefully crafted a nicely laid out and well-commented configuration file that you don't want to be overwritten, make a backup before using *Swat*. Another thing to note is that, although it is still part of the Samba suite, *Swat* isn't actively maintained any more and there have been discussions about dropping it completely. That said, it remains a popular choice for Samba administrators because it is useful as a learning tool and as a reminder of what the available options are and their default values.

Swat runs as a web service on port 901 of the Samba server. Point your web browser at, for example,

`http://myserver:901` to see *Swat*'s main page. You will need to have a login on the Samba server and use those credentials to log in to *Swat* (Samba credentials created with `smbpasswd` are not used).

The options available to you after logging in will depend on your ability to write to the Samba configuration file. The usual way to gain this right is to be a member of the `admin` group and for that group to have write access to the file. This will need to be preconfigured by a user with root privileges:

```
$ sudo usermod -a -G admin myuser
```

```
$ sudo chgrp admin /etc/samba/smb.conf
```

```
$ sudo chmod g+w /etc/samba/smb.conf
```

A user without write access can browse the Samba documentation, see server status, view the server configuration and change the Samba password for any user that they know the current password for. They can do this on remote Samba servers as well as the local one where *Swat* is running.

The Swat wizard

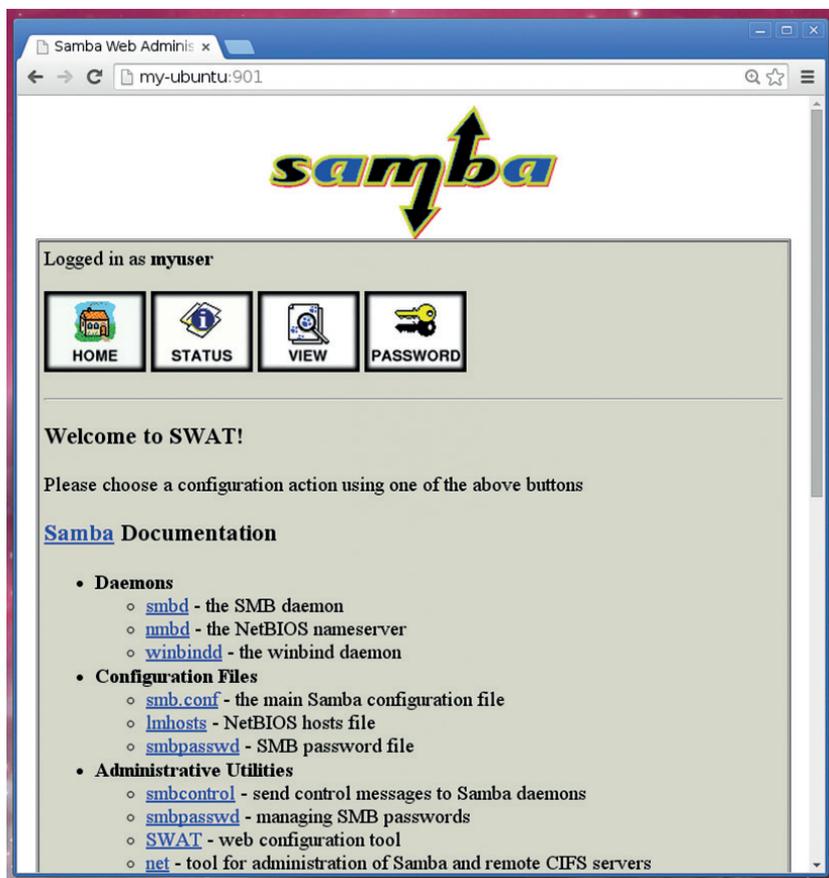
Users with write privileges also get access to the options used to configure Samba. There are screens to edit the global configuration, shares and printers. There is also a wizard to build configuration for you.

Aimed at "the Microsoft-knowledgeable network administrator", the wizard has two options: the first, 'Rewrite `smb.conf` file', rewrites `smb.conf` with the existing settings, ignoring any changes made but not written; the rewritten file will lack any comments or settings that were already Samba defaults. The documentation calls this a 'fully optimised format'. What this means is that it removes all unnecessary comments and any settings that are unnecessary because they are the defaults values anyway.

The other option that the wizard offers allows you to configure a new server, either standalone or as a domain member or controller (*Swat* doesn't offer options for the new Active Directory functionality introduced with version 4). You can select a WINS configuration and choose whether you would like to create per-user home directory shares. After selecting your desired options, click the commit button to write a new `smb.conf` file.

Using *Swat* can help you become familiar with the many available Samba configuration options because its pages display many of them along with their current or default values as well as hyperlinks that take you directly to the relevant part of the documentation. A button on each setting allows resetting to its default. Settings with their default values don't get written to the configuration file.

Because *Swat* exposes many of Samba's configuration options, it can be more overwhelming



for those with little knowledge of them. For basic configuration tasks, other tools may be more appropriate and one such tool is *system-config-samba*.

This is a Python GUI application from Red Hat that enables you to manage shares and users. It has some integrated help pages. You may find it in your distribution's repositories, or you can obtain the source from Red Hat (<http://bit.ly/sysconfsamba>).

\$ apt-get install system-config-samba

If *system-config-samba* is too basic, another option is *gadmin-samba*, part of the *GAdminTools* project. It needs to be run as root and also overwrites the **smb.conf** file, but it does warn about this when it starts. It contains lots of options and overwrites any existing configuration with a more comprehensive one that contains lots of settings, the reasons for which may not be clear if Samba configuration is new to you. You may find the resulting configuration is more complicated than you require (which may not be an issue if you only view it through a GUI application!).

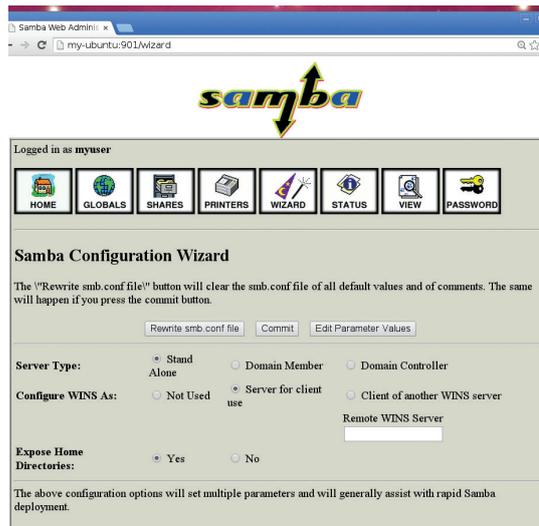
File access

Once Samba is configured, your main interaction with it will be for accessing files shared on the network by remote servers. Samba integrates well into desktop environments and allows shares to be browsed as easily as local filesystems.

Popular file managers like *Nautilus*, *Thunar* and *PCManFM* support virtual filesystems that can directly open remote Samba shares without the need for a separate step to mount them. They use a URI syntax to represent shares, and opening a share is as easy as using its URI, for example:

smb://myserver/myshare

What makes this possible are the virtual filesystem libraries like the Gnome Virtual Filesystem (*GVFS*) and the *KIO* library on KDE. Each file manager also allows you to create shortcuts to frequently used paths, for example by dragging them from the location bar into Places or adding bookmarks.



The *Swat* Wizard provides some basic options that can get you up and running quickly.

There are also GUI tools that can help if you have a large number of shares to manage. These often support multiple protocols, **smb://** being one of them. Two examples of these kind of tools are *Gigolo* and *PyNeighborhood*, available in many distros' repositories.

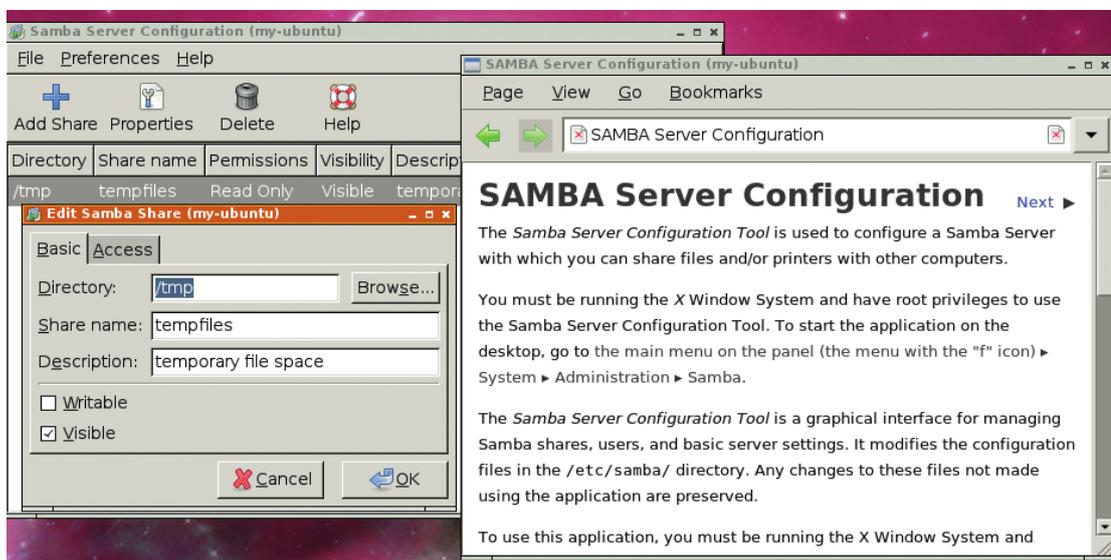
They offer a network browser to locate, select and mount shares. Alternatively, they allow remote server, user and share details to be specified manually. *Gigolo* supports multiple filesystem types, whereas *PyNeighborhood* specialises in SMB/CIFS browsing.

Most modern desktop environments have the integrated capability to browse Samba/Windows shares through their file managers and may include graphical administration tools as well. 📖

“Most desktop environments can browse Samba shares through their file managers.”

PRO TIP

After changing the configuration, Samba must reload it: the **smbcontrol all reload-config** command performs this task.



System-config-samba has the basic tools to manage shares and users.