BY: DINESH YADAV, LFY BUREAU

samba

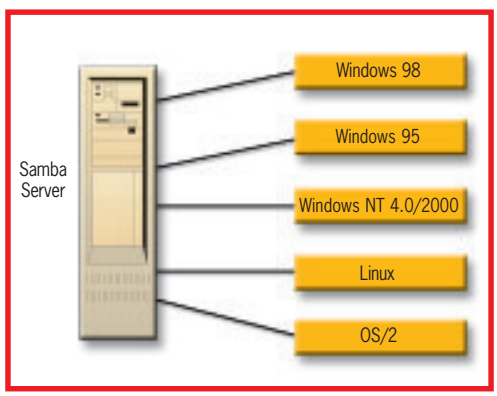
Looking for a server that can share files and printer services between different operating systems, aswell as connect Unix/Linux and Windows systems? The Samba server fits the bill. Here's how you could install and configure it.

Samba is an open source project developed by the Samba team, a loosely knit group of open source developers headed by Andrew Tridgell. It enables your Unix/Linux box to speak SMB (Server Message Block)—a protocol used to implement file-sharing and printer services among computers that have OS/2, Windows 9x and Windows NT as operating systems. It is basically used for connecting Unix/Linux and Windows systems. Samba can provide almost all functionalities that a Windows server offers. Once the server is set up on the network, it works in a very transparent fashion for end users. A great tool for network administrators, Samba may prove indispensable to mixed-platform environments.

UNDERSTANDING SAMBA AND THE SMB NETWORK

Networking with SMB involves several new concepts and is quite different from Linux/Unix networking. Here, let's start with the basics.

NetBIOS: NetBIOS (Network Basic Input/Output System) is a session layer communications service, developed by IBM in 1984. Initially, NetBEUI allowed each machine on a small LAN to claim a name that was not already in use on the network. Later, implementations of NetBIOS over Novell's IPX networking protocols also came into the picture. However, TCP/IP and UDP/IP networking protocols were the choice of the Internet community and hence implementing NetBIOS over these protocols became a necessity. The Internet Engineering Task Force (IETF) in 1987 released a series of standardisation documents outlining how NetBIOS would work over a TCP/UDP network.



Samba shares with all

Workgroup: SMB uses the concept of groups, with which machines can register themselves. Workgroup is a partition of machines on the same network.

Windows domain: This is a workgroup of SMB machines with one addition: a server acting as a domain controller. The domain controller is needed for a Windows domain.

Domain controller: The domain controller is the heart of the Windows domain, just as the NIS is the heart of the Unix network information service. Domain controllers have to perform a variety of functions and authentication is one of them.

Primary and backup domain controllers: The domain controller that is currently active on a domain is called the primary domain controller (PDC). There can be one or more backup domain controllers (BDCs) in the domain as well.

Local master and backup browser: The computer on the Windows workgroup (or domain), which maintains the list of machines that are accessible through the network, is called the local master browser and the list is called the browse list. The local master browser can have one or more backup browsers on the local subnet that will take over if the local master browser fails or becomes inaccessible.

Windows Internet name service (WINS): The Windows Internet name service (WINS) is Microsoft's implementation of a NetBIOS name server. The currently active WINS server is known as the primary WINS server. There can also be a secondary WINS server.

WHAT SAMBA CAN PROVIDE

The Samba server can provide many services; these can be:

- A file server and print server
- A primary domain controller
- A backup domain controller
- A domain master browser
- A local master browser but not a local backup browser
- A primary WINS server but not a secondary WINS server

COMPONENTS OF SAMBA

Samba is a collection of programs, which provide SMB services for Unix/Linux machines. A typical Samba package contains the following programs:

smbd: This is a server daemon, which manages the shared resources between the Samba server and its clients. It provides file, print and browser services to SMB clients. It is also responsible for user authentication, resource-locking and data-sharing through the SMB protocol.

nmbd: This is also a server daemon, which understands and can reply to the NetBIOS over the IP name service request. It participates in browsing protocols and can also be used as a WINS server (Windows Internet name server). Which means that it will act as a WINS database server, creating a database from the name registration requests that it receives and replying to queries from clients for these names. In addition, it can act as WINS proxy.

smbclient: This is a FTP-like tool that enables a client to access Samba shares.

smbtar: This is used to take a backup of Samba shares.

nmblookup: This is used for queering NetBIOS names and mapping them to IP addresses in a network over TCP/IP queries.

smbpasswd: This allows an administrator to change encrypted passwords used by Samba.

smbstatus: Is used for reporting the current network connections to the shares on a Samba server.

testparm: This validates the smb.conf, the Samba configuration file.

testprns: This is used to determine whether a printer name is valid for use in a service to be provided by Samba.

INSTALLING SAMBA

The easiest way to install Samba is to download the assembled package from http://www.samba.org and then use a package manager available in your system to install it. For example, in Red Hat distribution, you can use the following command to install it:

```
rpm -ivh <name of the package>
```

If you don't find the assembled package or you want to install Samba from its source code, download the source from http://www.samba.org, which would be a single compressed tar file. Then follow the instructions given below:

1. Unpack the archive with tar command, e.g., tar -xvf < name of the file >
2. Go to the newly created directory.
3. You will find a shell script called *configure*, which takes care of the machine-specific issues of building Samba. Execute this shell script to generate a *make* file. You can also set some global options by passing options on the command-line, e.g.,

```
./configure –with-smbmount
```

To see the complete list of global options, type the following command:

```
./configure – help.
```

4. Now compile the source code with the *make* command.
5. Install Samba using 'make install' command.
6. Finally, if you want to use SWAT (Samba Web Administration Tool), then add it to */etc/services* and */etc/inetd.conf* or */etc/xinetd.conf/swat* configuration files. SWAT basically provides a form-based editor in the Web browser for creating and modifying SMB configuration files and it runs as a daemon under *inetd* or *xinetd*

a) Add the following line to the end of the */etc/services* file:

```
swat  901/tcp
```

b) Add the following line to */etc/inetd.conf* file (you should also check the manual page of *inetd.conf* file to see the exact format of *inetd.conf* file):

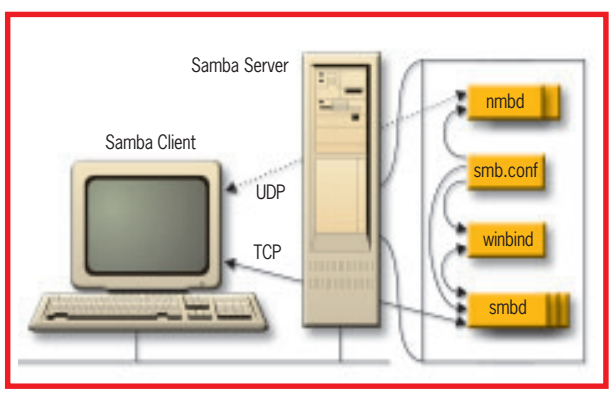
swat stream tcp nowait.400 root / usr/local/samba/bin/swat swat

or edit */etc/xinetd.conf/swat* and add the following lines to it.

```
service swat
        {
    port = 901
    socket_type = stream
    wait = no
    only_from = localhost
    user = root
    server = /usr/local/samba/bin/swat
    log_on_failure += USERID
    disable = no
        }
```

CONFIGURING SAMBA

The only configuration file for Samba is *smb.conf*. It determines which resources in the system can be restricted or shared with the outside world. It consists of sections and parameters—a section begins with its name and ends when the next section begins; sections contain parameters of



A Samba server at work

the form, name = value. Each new line or terminated line represents a comment, a section name or a parameter. Section and parameter names are not case-sensitive. Any line beginning with a semicolon or a hash character is ignored and a line ending in a '\' is continued on the next line. The values after the '=' sign in parameters are Boolean strings (yes/no, 0/1, true/false). Every section, except the [global] section, describes a shared resource. The name of the section is the name of the shared resource, and parameters within the section define the attributes of the share. There are three special sections, namely [global], [home] and [printer]. Sections other than these three are called ordinary sections and the following notes apply to them.

- A share consists of a directory to which access is being given, plus a description of the access rights, which are granted to the user of the service. Some housekeeping options are also applicable.
- Sections may be file share services or printable services and may be designated guest services, in which case no password is required to access them while for others, a password is required.

Global section: Parameters set in this section apply to the server as a whole, or are defaults for sections that do not specifically define certain items.

Home section: This section includes services connecting clients to their home directories, which can be created on the fly by the server. When the connection request is made, the existing sections are searched and if a match is found, it is used. Otherwise, the requested section name is treated as a user name and looked up in the local password file. If the name exists and the password, which has been provided is correct, a share is created by cloning the [home] section and some modifications are made to the newly created share:

- The share name is changed from homes to the located user name.
- If no path was given, the path is set to user's home directory.

Printer section: When the [printer] section is included, users can connect to any printer specified in the local host's *printcap* file. This section is quite similar to the [home] section. Suppose a connection request is made to a share, which is not present in *smb.conf* and its name can't be found in the password file. Samba checks to see if it is a printer share by reading the printer capabilities file to see if the share name appears there. If it does, Samba creates a share name after the printer.

Variables: There is a complete set of variables available in Samba for determining characteristics of the Samba server and the clients to which it connects. Each variable begins with a per cent sign, followed by a single uppercase or lowercase letter.

Configuration options: File configuration options in Samba are of two types—global and share. The options appear in the configuration file. Global options appear in the global section and nowhere else. But share options can appear in the specific shares section or global section.

STARTING SAMBA SERVER

Once you have configured the Samba server, you can start it by executing the following command:

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

If you don't find the smb script in */etc/rc.d/init.d/* directory, try to locate it. If you are unable to locate it, you can start Samba by executing the following commands:

```
smbd -D
nmbd -D
```

General services that Samba can provide:

Sharing your Linux/Unix directory with Windows

This is quite simple. Suppose you want to share a directory with the public, add the following lines in the *smb.conf* file.

```
[public]
  comment=Public Directory
  path=/home/foreveryone
  public=yes
  writable=yes
```

Now you can start or restart the Samba server and browse the */home/foreveryone* directory from Windows machines.

ACCESSING A SMB SHARE FROM LINUX/UNIX MACHINES

You can use the smbclient program that comes with the Samba suite to access a SMB share, irrespective of whether the server is a Windows machine or a Samba server. If you want to see which shares are available on a host, say, Gaurav, then you can type the following command:

```
smbclient -L Gaurav
```

And if you want to access a particular share available on Gaurav, type the following command:

```
smbclient  \\\\Gaurav\\sharename passwd
```

where sharename is the name of the particular share and passwd argument contains the password required to access that share. You will then see the *smbclient* prompt.

```
smb:\>
```

You can now use the available commands, and if you have ever used FTP, you'll find all this easy, as smbclient is quite similar to FTP. Otherwise you'll need to check its man page. You can also use the smbfs package that contains the *smbmount* and *smbumount* program, which are similar to *mount* and *umount* for smb shares. For these programs to work correctly you must have smbfs support compiled into your kernel.

SHARING YOUR UNIX/LINUX PRINTERS WITH WINDOWS

If your printer is installed on your Unix/Linux machine, then setting up an SMB share of the printer is quite simple. You

only have to add the printing configuration option to your *smb.conf* file, e.g.,

```
[global]
   printing=bsd
   printcap name = /etc/printcap
   load printers = yes
[printers]
   comment = All Printers
   security = server
   path = /var/spool/lpd/lp
   browseable = no
   printable = yes
   public = yes
   writable = no
   create mode = 0700
[Ink jet]
   security = server
   path = /var/spool/lpd/lp
   printer name = lp
         writable = yes
   public = yes
   printable = yes
   print command = lpr -r -h -P %p %s
```

Once you have specified *printcap = /etc/printcap* and *load printers = yes* in the [global] section, then all printers that are present in the /etc/printcap file will be loaded by defaults. The [printers] section specifies options for the printers that you want to define explicitly.

SHARING YOUR WINDOWS PRINTER WITH UNIX/LINUX

Your Windows printers can also be accessed from the Samba client (here it's the Unix/Linux machine). The *smbprint* tool that comes with the Samba suite can be used to print jobs to Windows-based printers. However, to use this, you need to set up the printer as a shared resource on the Windows machine to which the printer is connected. Once you have done this, Samba can make it available to all other clients in the workgroup. The Samba distribution provides two simple scripts: *smbprint* for BSD-style printers and *smbprint.sysv* for System V printers.

SETTINGS FOR BSD PRINTERS

There are two steps here. You need to have a BSD Unix recognise a remote printer.

First edit your */etc/printcap* file and add an entry for the remote printer. The input filter (if) entry needs to be printed on the *smbprint* program when the machine is on Windows. After that, you should create a configuration file in the spool directory, which you would have specified with *sd* parameter in */etc/printcap* file. The name of this file should be .config and should contain the following information:

- The NetBIOS name of the Windows machine with the printer
- The service name that represents the printer
- The password used to access that service

You can specify these details in .config file in the following manner:

```
server = Gaurav
service = hp
password = " "
```

Now start or restart your Samba server.

SETTINGS FOR SYSTEM V PRINTERS

Locate your *smbprint.sysv* script, which comes with Samba distribution, edit it and change server, service, and password parameters to match the NetBIOS machine, its shared printer service, and its password, respectively. Now run the following commands, which create a reference for the printer in the printer capabilities file. Note that the new Unix printer entry hp_ printer is named:

```
lpadmin -p hp_printer -v /dev/null -i ./
smbprint.sysv
enable hp_printer
accept hp_printer
```

Now start or restart your Samba server.

TROUBLESHOOTING SAMBA

Although Samba is very robust, you may face occasional problems. You can use the following tools to troubleshoot Samba server.

Samba logs: Whenever you face any problem related to Samba, first check Samba's logs files. You can set the logging level of Samba server—substitution variables that allow you to isolate individual logs for each machine, share, or combination thereof—to log as little or as much as you want. Generally, logs are placed in */var/log/samba/smbd.log* and*/var/log/samba/nmbd.log*. You can change the location of these files by using the *log file* configuration option in smb.conf. You can also specify a log directory to use with flag-l on command line, e.g., smbd -l */usr/local/samba/*. There are different levels of logging but level 3 is sufficient for most of the Samba administrators. For 'no logging,' you can set the level to 0. You can also increase or decrease the level of logging by one, by sending signal SIGUSR1 or SIGUSR2 to smbd process.

Samba test utilities: A rigorous set of tests that exercise the major parts of Samba is described in various files in the */docs/textdocs* directory of the Samba distribution kit, starting with *DIAGNOSIS.TXT*, which covers installation and reconfiguration diagnosis. The other files in the */docs* subdirectories address specific problems (such as Windows NT clients) and instruct you how to troubleshoot problems.

Linux utilities: Two commands, *tcpdump* and *strace*, can help in debugging problems related to the Samba server. The *strace* command displays each operating system's function call as it is executed for a particular program such as the Samba server, and it will often pinpoint the exact call that is causing the difficulty. The *tcpdump* command allows you to monitor network traffic in realtime, using which you can examine all conversations between the client and the server, including SMB and NMB broadcast messages. You can use its output to get a general idea of what the server and client are attempting to accomplish. LFY