Mastering Linux, part 16

Need to share files between Linux and Windows systems? *Jarrod Spiga* **explains how you can do it by putting SMB and NFS protocols to work.**

JARROD SPIGA

Jarrod Spiga is an infrastructure and network engineer who specialises in getting Linux and Windows systems happily coexisting with each other. He currently holds CCNA and numerous MCP certifications, and dabbles in web application development in his spare time.

SKILL LEVEL

● Advanced

TIME TO COMPLETE

● 3 hours

REQUIREMENTS

● An installation of Linux (Fedora Core 4 and Red Hat Enterprise Linux ES 3.0 were used in the writing of this Workshop).

IN THIS SERIES

One of the main reasons why people connect systems on a LAN, besides internet connectivity, is to share files and copy data from one PC to another. Linux can share such data with Windows PCs by utilising the Server Message Block (SMB) protocol with other Linux or Unix systems using the Network File System (NFS) protocol. However, some configuration is required to use either of these protocols.

SHARING WITH WINDOWS

Very little configuration is involved on a native Windows network. Once TCP/IP is configured, the NetBIOS broadcasts are combined with the SMB protocol and an optional WINS server to discover all compatible hosts on most networks.

Most Linux systems aren't configured to discover Windows shares out-of-the-box, and you'll need to use Samba to communicate with Windows hosts via SMB. Samba is typically configured from within a text file, but it can just as easily be managed via the Samba Web Administration Tool (SWAT).

The first step in getting SMB file sharing running is to ensure that the Samba service is installed. You can install it by opening the Add and Remove Software tool and installing the Windows File Server option, but it's a better idea to download the latest RPM packages for Samba — so you're not installing the outdated version included with your installation media.

INSTALLING SAMBA AND SWAT

❶ ❷ If you're using a distribution that supports yum, the easiest way to install Samba is to bring up a command line, log on as the root user and run:

```
yum install samba
yum install samba-swat
```

Downloading the samba-swat package is optional (but recommended for this Workshop). If you don't have a yum-capable distro, the latest version of the Samba binaries can be downloaded from **http://au.samba.org/samba/ftp/Binary_Packages** (you should be able to navigate to the repository for your distro from this URL). Once you've downloaded all four packages, bring up a command prompt, log on as the root user and install them by entering:

```
rpm –ivh samba*.rpm
```

Once Samba is installed, you may wish to configure the services so they start when the system is booted (or, in the case of SWAT, when X-Windows is loaded). Enter:

```
/sbin/chkconfig --level 345 smb on
/sbin/chkconfig --level 5 swat on
```

❸ Samba's configuration file is usually located at /etc/smb/smb.conf — it contains many comments, so most users will be able to configure Samba to operate properly. However, SWAT is a simpler tool to use to generate Samba configurations.

1

Not Pepsi Samba, but still yum: the Yellow-dog Updated Modified application makes installing the latest version of Samba trivial.

2

SWAT team: installing the optional Samba Web Administration Tool makes configuring Samba less of a hit-or-miss affair.

3
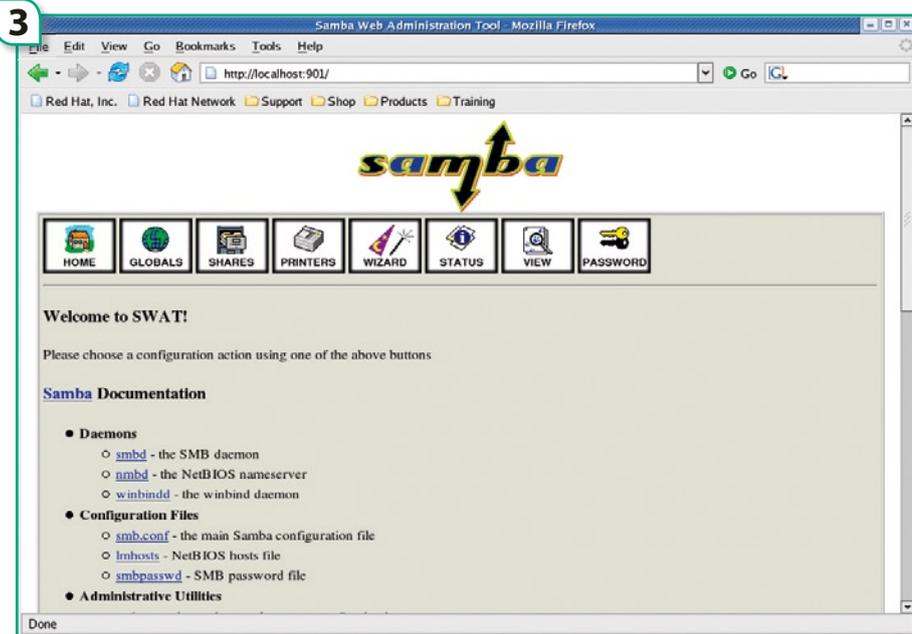
The main SWAT page provides links to detailed documentation covering all facets of how to use Samba.

The SWAT configuration page can be brought up by loading a web browser on your Linux system and navigating to **http://localhost:901**. Before the page displays, an authentication prompt will appear — root user credentials are required to make changes to the Samba configuration.

Once SWAT loads up, the home page provides links to all of Samba's documentation. You could start configuring shares straight away, but it's a better idea to configure some global settings first. To get started, hit the Globals icon toward the top of the page.

❹ These settings control the SMB server on your Linux system — which in turn affect how the Linux system appears to the Windows hosts on a network. You'll need to configure the following fields in the global configuration page:

Workgroup: the workgroup that your Linux system will appear under when a Windows client browses through Network Neighbourhood or My Network Places.

NetBIOS name: the name you want your Linux system to be known as on the network. This is typically the same as your hostname.

Interfaces: interfaces Samba should bind to. In most cases, you'll want Samba running all local Ethernet interfaces (eg. eth0).

Security: the security level used on the Samba Server. If set to USER, clients need to supply authentication credentials in order to access local shares. If set to SHARE, anonymous access is granted. If set to one of the remaining three options, authentication requests are passed to another server (be it an Active Directory, NT domain or another Samba server) before access is provided.

Once you've configured these settings, along with any other settings that you may wish to

customise, scroll to the top of the page and click on the Commit Changes button.

CREATING SHARES

❺ SMB shares on your Linux system can be created by hitting the Shares button near the top of the SWAT Web applet. In the Create Share text field, type in the name that you want the new share to be known as, then click on the Create Share button.

❻ Now the properties page for the share will appear. At minimum, you should configure the following fields for each share you create.

Comment: a comment field which appears next to the share name from most Windows clients. This field is generally used to describe the contents of the share.

Path: the full path to the directory on the local file system you wish to share.

Read only: if set to yes, clients won't be able to write to the share.

Guest OK: if set to yes, clients won't necessarily have to supply authentication credentials in order to gain access. If set to no, clients must provide authentication.

Once you've modified all of these fields (plus any others you want to customise), scroll back up to the top of the page and click the Commit Changes button.

When creating shares, you need to be mindful of permissions. For instance, if guest access is permitted on a share, the directory must be readable (and perhaps writable) by everyone.

The process that needs to be followed to edit an existing share is just as simple. After clicking on the Shares graphic at the top of the screen, select the share you wish to edit from the Choose Share drop-down list and click the button with the same name.

Once you've finished working with your shares, start the Samba service from the command line by typing:

```
/sbin/service smb start
```

FIREWALL CONSIDERATIONS

Like all other network services, you'll need to adjust the firewall config for your SMB shares to become visible to other hosts on the network.

For your shares to be seen, you'll need to allow inbound (ingress) connections for both TCP and UDP connections on ports 137 (NetBIOS Name Service), 138 (NetBIOS Datagram Service) and 139 (SMB/NetBIOS Session Service). And if you want SWAT to be accessible from other systems on your network, you'll need to allow inbound TCP connections on port 901.

4
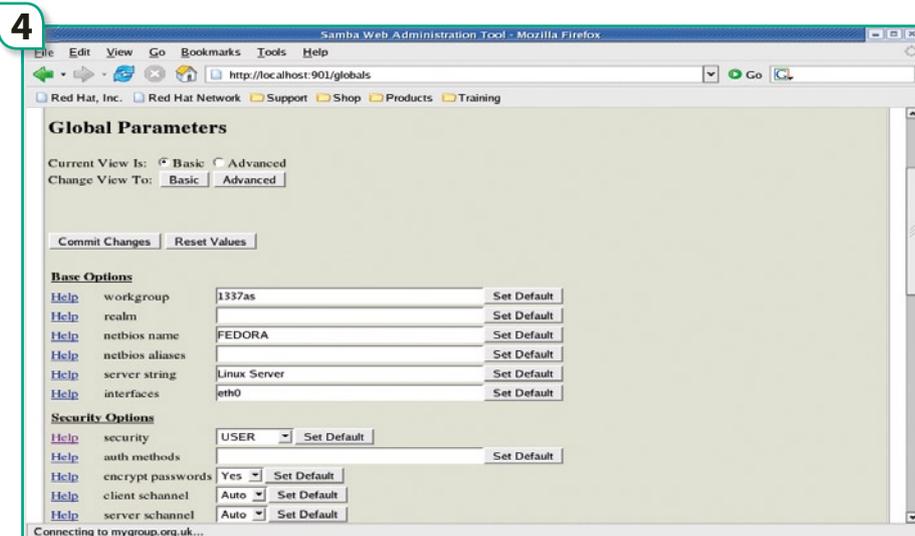
Think global: before enabling Samba, ensure that your global configuration is correct.
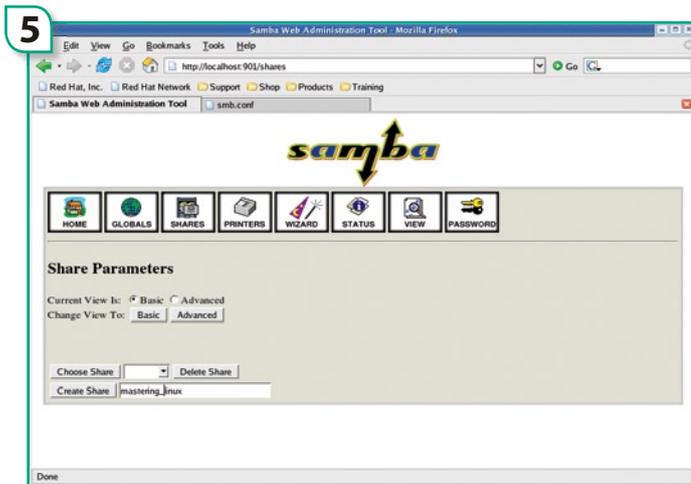
5

My mother said it's nice to share: to create a share, enter the share name and click Create Share.
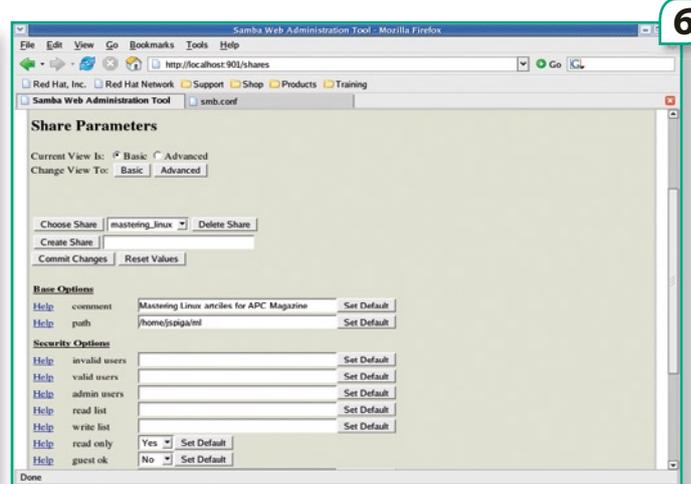
6

A detailed sharing arrangement: the Share Parameters screen provides you with various access control mechanisms for each share.

"NFS IS EXTREMELY POWERFUL AND CAN BE OPTIMISED AND SECURED — SO EVEN WAN IS FAST AND SAFE

AND FOR THOSE OTHER LINUX SYSTEMS…

Not only does Samba provide the ability for file sharing between Linux and Windows systems, it also provides a mechanism for Linux systems to share data with other Linux systems. However, the use of the SMB protocol is not the de facto standard for file sharing in the Unix world. That title belongs to the Network File System (NFS) service. NFS is so widely used because it's a core part of most Linux and Unix distributions, and configuration is about all that's required for an NFS to start operating.

CONFIGURING THE NFS SERVER

The NFS service is configured based on the contents of the /etc/exports file on most Linux and Unix systems. Each line of this file could be thought to represent one share on the local system. The actual format of each line is:

```
/path/to/data hostname.domain.com(access)
```

where /path/to/data is the local system path being shared, hostname.domain.com is the hostname or qualified domain name of the system permitted access to this local system path, and access is the level of access given to that host (ro is for read-only access, while rw provides read-and-write access).

For instance, the following line would configure NFS with a share similar to what is offered to Windows clients on the network:

```
/home/jspiga/mastering_linux *.1337.as(ro)
```

In the above example, all hosts from the 1337.as networks would be granted read-only access to all of the work I've done for the Mastering Linux series. You can replace that string with the name of a specific host, or even an IP address and subnet mask combination, eg:

```
/home/jspiga/mastering_linux 192.168.1.0/25
5.255.255.0(ro)
```

Once you've configured the /etc/exports file, it's time to start the NFS service. This is done in the same manner as most services are started:

```
/sbin/service nfs start
```

If you'd like to configure your system so NFS starts on boot, use the chkconfig command:

```
/sbin/chkconfig --level 345 nfs on
```

Finally, any firewall you're running will also need reconfiguring to permit access. For NFS to work, you'll need to allow inbound TCP and UDP connections on ports 111, 369 and 2049.

THE NFS CLIENT

Sharing locations on your system is one half of the process, accessing these shares from another system is the other. NFS shares need to be mounted in the same manner that physical drives are mounted on the root file system.

Mounting an NFS on an ad hoc basis is as simple as creating the directory you wish to use as a mount point (as explained in part 13 of this series, see *APC* December 2005, page 128) and then issuing the appropriate mount command. Following on from the previous examples:

```
mkdir /mnt/ml

mount fedora.1337.as:/home/jspiga/
mastering_linux
```

Alternatively, an NFS can be mounted every time your system starts by editing the /etc/fstab file. To mount the above NFS, add (noting tabs):

```
fedora.1337.as:/home/jspiga/mastering_
linux      /mnt/ml   nfs      ro       0 0
```

FURTHER INFORMATION

Due to space constraints, the intricacies of both methods of sharing files haven't been discussed, as they are considerably more complex than what has been outlined in this Workshop. But the basics have been shown.

For instance, Samba can also be configured to operate as a "free-as-in-beer" Windows NT domain controller, and can even be used to join your Linux system to Active Directory-enabled Windows domains. When used in conjunction with WinBind and Kerberos software on your Linux system, you can even utilise many of the advantages of Windows domains on your Linux system.

Samba can also be used to share printers, whether the print device is attached or virtual. And by extending the concept a little further, you can even set virtual printers to generate PDF files via a Linux printer shared in Samba.

The official Samba HOWTO (**http://us1. samba.org/samba/docs/man/Samba-HOWTO-Collection/**) describes how to do these things and more.

Similarly, NFS is extremely powerful and can be optimised and secured — so even sharing data over WAN connections is fast and safe. The NFS HOWTO can be found at **www.tldp.org/ HOWTO/NFS-HOWTO/index.html**.

NEXT MONTH

● Next time, Mastering Linux will discuss a number of internet services in more detail, including how to configure your Linux system as a SMTP/POP3/IMAP mail server.