How to Network Red Hat Linux and Microsoft Windows

by Larry F. Byard

Dux Computer Digest

http://duxcw.com/

CONTENTS

This article will show you step−by−step how to setup a computer running Red Hat Linux version 7.3 in a network and computers running various versions of Microsoft Windows. The Windows computers will be able to share files and printers with the Linux computer and vice versa, and it will be possible to move files and folders back and fourth between the computers.

INTRODUCTION. Linux is a Unix−like operating system originally created by Linus Torvalds in 1991. Over time individuals and companies began distributing Linux with their own suite of accompanying modifications and software. Presently, the distribution of Linux has become a multi−million (if not, billion) dollar business. However, Linux was developed under a GNU general public license and is free to anyone who wants it. Various Linux distribution packages, with all sorts of quite respectable and powerful software, are available as free downloads from numerous web sites on the Internet. These downloads are huge (one really needs a reliable, high−speed cable or DSL broadband Internet connection to download them) and usually consist of CD−ROM CD images from which CDs can be made for installation with a CD−RW drive. Distributions are also available on CD with documentation and various sorts of support at modest prices from retail outlets such as Walmart.

Linux will run most of the software you see on the Internet (much of the Internet is run on Linux) and much more. It can act as either a workstation, server, or both on a Local Area Network (LAN) and provide most of the services available from other, often expensive server software packages. It is not as user−friendly to most people as some other systems such as Microsoft Windows.

Red Hat (RH) is the largest Linux distributor and various Red Hat Distributions are running on about 75% of all of the Linux−based computers in the world. That does not necessarily mean it is the best distribution or the best one for a particular application, or that Linux is better than other free unix−based packages such as FreeBSD.

Many Linux distributions include an open source program called Samba. Samba is also available as a free download on the Internet. The Samba web site describes Samba as a

> "Protocol by which a lot of PC−related machines share files and printers and other information such as lists of available files and printers. Operating systems that support this natively include Windows NT, OS/2, and Linux and add on packages that achieve the same thing are available for DOS, Windows, VMS, Unix of all kinds, MVS, and more. Apple Macs and some Web Browsers can speak this protocol as well. Alternatives to SMB include Netware, NFS, Appletalk, Banyan Vines, Decnet etc; many of these have advantages but none are both public specifications and widely implemented in desktop machines by default."

In short, Samba software allows you to integrate a Linux computer into a Microsoft Windows network. It can also make Linux a dedicated server and network domain controller.

OUR NETWORK. Our home/office/shop network consists of several computers at any one time in my Office/Shop and two computers used my Wife and Son, which are located in their respective offices. All are networked on a 100BASE–TX Ethernet local area network (LAN) with an SMC7008BR combination broadband router/seven–port Ethernet switch. The router also provides shared access to the Internet via a cable MODEM. The router has a DHCP server and is used by all of the PCs used as workstations to manage IP addresses. The PCs on the network are running Windows 98, Me, and 2000 Pro operating systems. Other computers with Window NT Server (old server used for support) and XP operating systems are occasionally introduced into the system. The network is usually configured as a peer–to–peer Windows network with workgroup name of WORKGROUP. The Linux server is built from used parts (one does not need a speed demon for a server on a small LAN; save that for the user PCs) and was added to the network primarily as a learning testbed, to see how Linux stacks–up as a small business/home office workgroup server as compared to various Windows versions and Novell, and as a development system for our web sites, which are running on a remote FreeBSD unix system maintained by a hosting service. The web sites employ an Apache web server and run several large scripts that use php and MySQL. Also, my Son might try to "barrow" the Linux machine (when I'm not looking) for a game server at a LAN party.
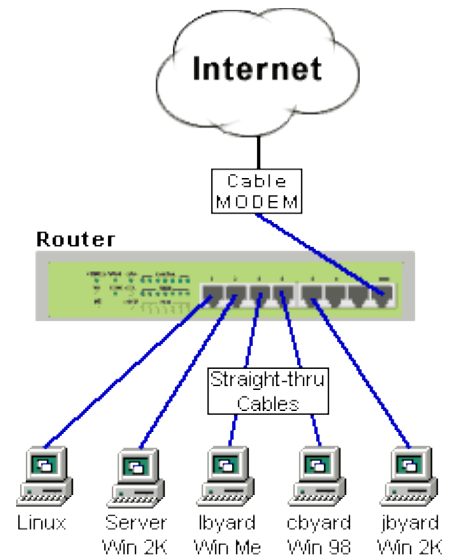
ROUTER CONFIGURATION. This configuration can be done after the Linux install. Many LAN/routers use 192.168.100.x or 192.168.0.x. Ending the network IP with 0 and the Broadcast with 255 is a common practice. The Gateway and Primary DNS were assigned the IP of my router. It is the same IP that one uses to view the router's HTML interface with a browser. The router has a DNS proxy server. These settings will be different if the firewall/router is installed and Linux host is going to perform the router NAT function. The router is configured like so:

The last 6 digits of the MAC address were changed to X's in the picture so as not to make them public for possible(?) security reasons. It is important that the Linux host static IP address does not fall within the scope of the DHCP server and that it be on the same network segment (192.168.123.*) as the router and PC's. On can determine the Ethernet adapter's MAC in RH 7.3 by opening a terminal window (see below) and typing ifconfig. HWaddr is the MAC address.

Linux could be setup with its firewall, a second network adapter, and an Ethernet switch or hub to perform the broadband router function, but a router does it better for about the same cost and uses less energy. Linux could also be setup in a similar manner, but with a MODEM instead of a second network adapter, to share a dial–up Internet connection. Samba could be run with a dynamic IP address obtained from the router.

SECURITY AND BASIC NETWORK CONFIGURATION. As all users of the system are trusted, security is not a problem (not really required). However, I did not want novice users to "accidentally" access those parts of the Linux host that would normally be managed by root. Simplicity is a goal. At this point, I do not plan to attach a printer to the Linux computer or print to a network printer on another computer from the Linux computer. I decided to have one directory that could be shared by all users and one directory for each user that only that user could access. User directories would be restricted with a password. This configuration is pretty much what Red Hat has outlined in the RH Customization Guide. This article elaborates on that guidance and stresses salient points that might other wise be overlooked. So, here is Samba simplified:

INSTALLATION STEPS (Method 1)

1. Red Hat Linux 7.3 Installation. I did a custom install.   Relevant parts of the install follow.

Password options. I don't entirely understand all of the RH password gibberish, but here is what I did.

- ◆ Enable MD5 password: checked
- ◆ Enable shadow passwords: checked
- ◆ SMB tab:
    - ◆ SMB Server: Samba Server
    - ◆ SMB Workgroup: WORKGROUP

The gnome GUI (graphics user interface) was installed.

The smb (Samba) server was installed.

The RH firewall was turned–off. The router performs this function for our LAN and experience shows that firewalls on a LAN behind a router can cause LAN problems.

A Linksys LNE100TX 10/100 PCI Ethernet Adapter was installed in the computer, the RH install detected it and automatically installed the driver. The network configuration was entered as follows:

- Configure using DHCP: unchecked
- Activate on boot: checked
- IP address: 192.168.123.1
- Netmask: 255.255.255.0
- Network: 192.168.123.0
- Broadcast: 192.168.123.255
- Hostname: linux
- Gateway: 192.168.123.254
- Primary DNS: Gateway: 192.168.123.254

The following additional user accounts were created:

- lbyard (me)
- cbyard (my Wife)
- jbyard (my Son)
- pcguest (shared LAN account)
- Password authentication

   This can done later with the User Manager in gnome (see below).

When RH first boots–up **login as root**.

2. Setup Linux to use the LAN and Internet. In gnome double–click the **Start Here** icon at the top–left corner of the screen. Double–click **System Settings** in the resulting window, double–click **Network Configuration** in the next window. You should see **eth0**, the Linux device name for your Ethernet adapter. Click the **Host** tab and verify/enter the IP address, Hostname, and any aliases: E.g.:

192.168.123.1
linux. WORKGROUP.lan
linux

On a private LAN you can use anything you want for it within reason. WORKGROUP.LAN corresponds to a domain name. We plan to add Apache–based, virtual, intranet development web servers that will be patterned in the same manner: e.g., duxcw.WORKGROUP.lan with an IP of 192.168.123.2.

Click the **DNS** tab and enter/verify the DNS (Domain Name Service) server IP address and the domain name. The IP address corresponds to the router, which is running a DNS proxy server like most broadband routers. You could enter your Internet Service Provider's (ISP) Primary and Secondary DNS server IPs if desired. The Hostname has no significance in this configuration and may be deleted if desired.
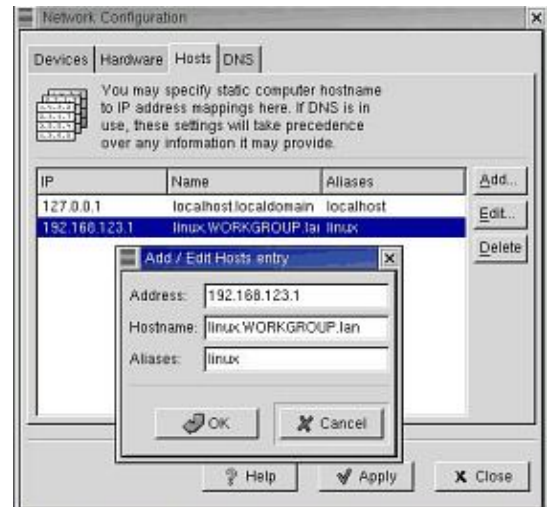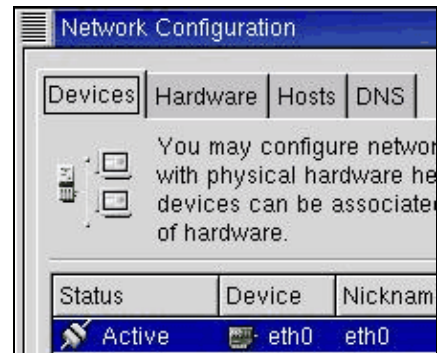
3. Create the Linux user accounts if you haven't already done so. In gnome, click the gnome **Start** (I'll call it "Start"), which is the G–shaped footprint at the bottom–left of the screen, **Programs**, **System**, **User Manager**. Account names are case sensitive, should match the Windows login name for Windows to use Samba with the least amount of hassle. Passwords must contain at least seven characters and are case sensitive. Larry's rules for Windows names, will probably keep you out of trouble. Unix does like lower case, however. A common practice for unix names is to use the user's first initial and last name and run them together, all lower–case. Of course, root already exists, but is not listed and cannot be entered again with the User Manager.
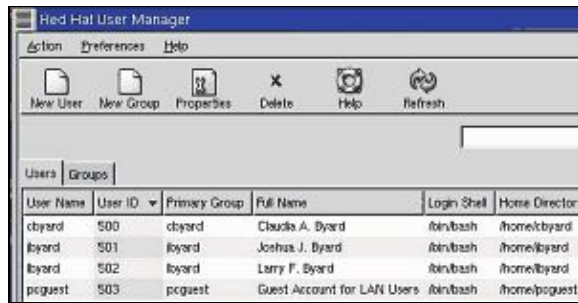
4. Create a Samba password file based on the existing Linux password file (/etc/passwd). That is, Samba has its own user/password file (/etc/samba/smbpasswd) in addition to the one required for Linux. Yes, you have to do the following or restricted connections to user home directories won't work. Click the **Terminal** icon on the task bar at the bottom–left of the screen to open the Terminal emulation program. Type the following at the prompt:

cat /etc/passwd | mksmbpasswd.sh > /etc/samba/smbpasswd

The preceding command does not copy the passwords. Manually set each Samba user's (including pcguest) password with:

smbpasswd *usename*

E.g., smbpasswd lbyard

And enter the password in the resulting window. Don't forget root. That also activates the user's Samba account.

Type the following:

**chmod 600 /etc/samba/smbpasswd**

... to restrict read/write permission to root.

Once you have created an smbpassword file, use the smbpasswd –a <username> *command to add any new users to the file. The command will not work until after you have created the smbpasswd file.*

5. Make a shared folder for the LAN users. In gnome, click the **root's Home** icon in the upper–left corner of the gnome desktop (or click Start, Programs, Applications, Nautilus) to bring–up gnome's equivalent of the Windows Explorer file manager. Click the **up button** in the Nautilus menu to go to the root (/) directory/folder. Double–click the "**home**" folder to open it. Click **File** in the menu, select **New Folder**, and enter "**shared**" as the folder name. Click the **Permissions** tab, change the **File Owner** to pcguest, the **File Group** to pcguest, and **check all of the boxes** to give the file a chmod "Number View" of **777**.
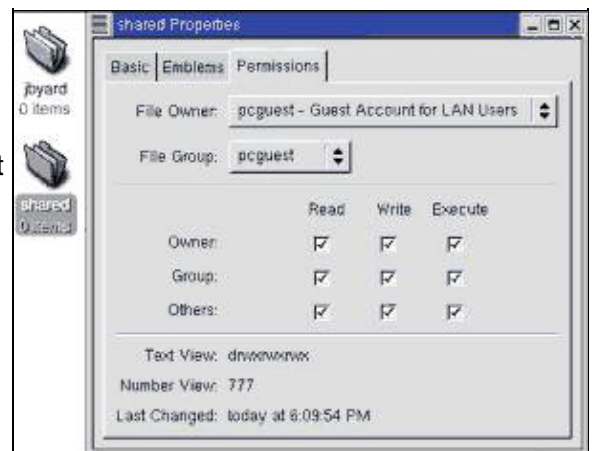
Actually, the File Owner and Group do not have to be changed. You can call this file whatever you want and put it elsewhere if desired. You could use /home/pcquest for the shared directory. Or, omit it altogether if you don't want a shared folder/directory.

6. Edit the Samba configuration file. While still in Nautilus navigate to the following folder: **/etc/samb**. Click on **smb.conf** to highlight it. In the menu click **Edit**, **Copy File**, **Edit**, **Paste File**. That will make a backup copy of the configuration file in the same folder as the original. Click **Start**, **Programs**, **Applications**, **qedit**, and **Open /etc/samba/smb.conf**. This is what our smb.conf file looks like:

```
[global]
workgroup = WORKGROUP
server string = Samba Server
hosts allow = 192.168.123. 127.
guest account = pcguest
security = share
encrypt passwords = yes
smb passwd file = /etc/samba/smbpasswd
unix password sync = Yes
passwd program = /usr/bin/passwd %u
passwd chat = *New*password* %n\n *Retype*new*password* %n\n
*passwd:*all*authentication*tokens*updated*successfully*
pam password change = yes
obey pam restrictions = yes
socket options = TCP_NODELAY SO_RCVBUF=8192 SO_SNDBUF=8192
[shared]
comment = Shared read/write directory for all LAN users and guests
path = /home/shared
public = yes
only guest = yes
writable = yes
printable = no
[homes]
comment = Home directories
browseable = no
writable = yes
valid users = %S
create mode = 0664
directory mode = 0775
map to guest = bad user
```

Some of the longer lines above are wrapping. They should paste OK, however. Every line except those in square brackets has an = in it.

To duplicate it, enter the lines in ours and delete the rest (you have a backup) or place a ; in front any other lines in yours that have not been made into comments. Or, if you are looking at this page with a browser in Linux (that is possible at this point in my configuration/network) copy and paste my lines into your file or make a new smb.conf file with them. I use Mozilla (**Start**, **Programs**, **Internet**, **Mozilla**) for occasional Internet browsing from RH for this installation because it is already installed by default and ready to use, and the version on Netscape that comes with RH 7.3 is not configured and ready to use, and is an ancient version (the Mozilla is old, as well).

7. Start–up Samba. Click **Start**, **Programs**, **System**, **Service Configuration**, scroll down to **smb**, and **check** the checkbox next to it. In the Service Configuration menu, click **File** and select **Save Changes** so the Samba daemons (programs that run in the background to handle routine tasks/implement functions) will be automatically started when Linux boots. Then click the **Start** icon to start Samba the first time (you may still have to reboot to get it going). The network should be working within a few minutes, if not sooner. It might take some time for Linux to show–up in all of the network Browse Lists and you may have to restart Windows on the workstations.

8. Check Networking from Linux. To connect to Windows computers from **Nautilus** in Linux type **smb:** in the location box. That should display the **WORKGROUP** icon. Double–clicking it will display the Windows Computers that are running, etc.

9. Check the Network from Windows. I am assuming that you already have an operational Windows network. If not, our various networking How to articles and FAQs should help with that configuration and with troubleshooting. Additionally questions and problems can be posted in our Forums. As an example, my Windows Me network configuration (Start, Settings, Control Panel, Network...) is shown to the right. The **TCP/IP properties** are set to **obtain the IP address automatically** form the router. The Log on to Windows NT domain checkbox in the **Client for Microsoft Networks** properties is **unchecked**. Be sure the log on name for Windows/Client for Windows Networks matches the Linux/Samba account name.

I did have a problem with the Windows Me computer. When trying to open Entire Network in the Windows Explorer, an error message appeared stating that it could not access the network and to Wait. The fix was to uninstall everything network on the Me computer, restart Windows, reinstall the network, and restart Windows again. This problem occurred during both of the installs of Samba performed for this article. The other computers posed no problems.
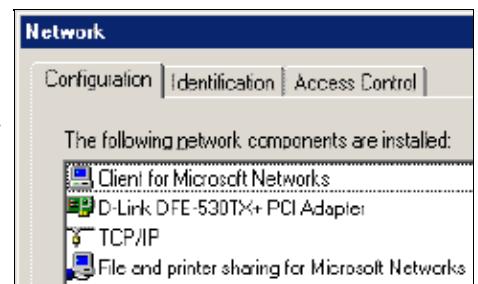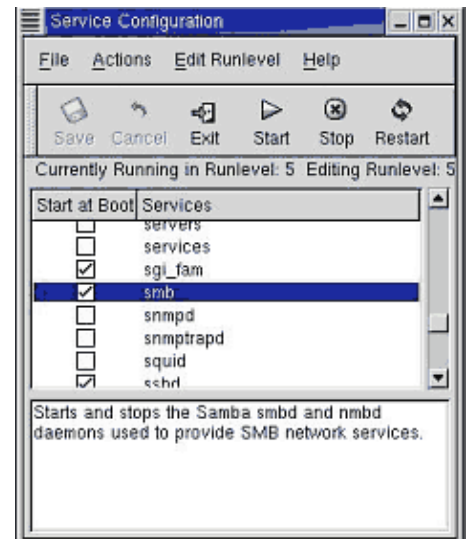
In the Windows Explorer the Network Neighborhood/My Network Places should look something like the screenshot to the right. The other user home directories will not appear. Clicking on the home directory (lbyard in this case) a screen will appear prompting for the Samba password. Once entered Samba/Windows will remember who you are and will not prompt for the password again unless you login into Windows/The Client for Microsoft Networks with a different user name. Note the Shortcut that Samba has created in the right Window pane. You can drag or copy that shortcut to your desktop or Quick Start bar and use it to fire−up a window containing your home directory contents. It should appear almost instantly. Linux folders can be mapped to drive letters in Windows just like any other network shares. See MAP THOSE DRIVES for instructions.
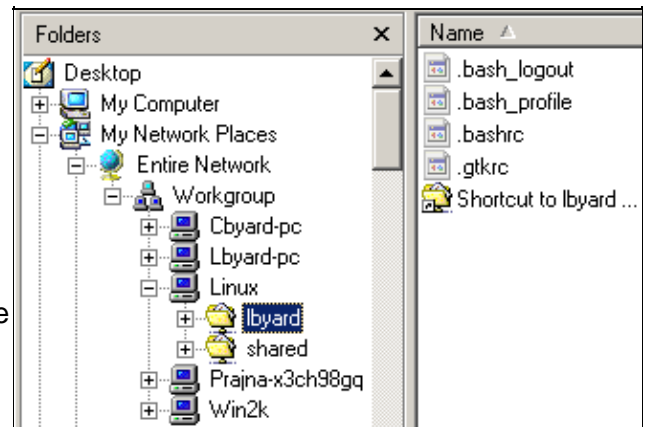
INTALLATION (Method 2). I am not going to go into some of the details of this method as they are covered in Method 1. Instead of installing Samba from the RH CDs during the installation process, this method installs a download of an RPM (Red Hat Package Manager) version of Samba. One can find RPM's on numerous Internet web sites. A couple you may want to try are:

> http://www.rpmfind.com/

> http://rpmfind.net/linux/RPM/

I first tried the one that was available on Samba and it crashed consistently. It was version 2.2.5−1 at http://us2.samba.org/samba/ftp/Binary_Packages/redhat/RPMS/7.3/. They may have fixed it by the time you get ready to do it.

I then used version samba−2.2.5−2.i386.rpm from http://www.firerun.net/pub/i386/samba/ and it installed without a problem and is what I am running now (9/7/02).

1. If you haven't already, **Install RH** as before, except do not install Samba. Don't worry if you have already installed it. The RPM Manager will allow a **full install** of Samba or an **upgrade**. In fact, you could do method 1 to get Samba going and simply upgrade it later. I strongly recommended that you **backup the files in /etc/samba if you are going to do an upgrade** of an operating Samba network. The RPM should automatically back−them−up also.

2. Login as root, connect to the download site, **and download the RPM**. As I recall the file is about 10 Mbytes in size. If you haven't changed directory since logging in Linux, the download will be placed in /root and that is where you want it

10

3. Once the download is completed, use the gnome RPM to install it (**Start**, **Programs**, **System**, **GnoRPM**). In the gnome RPM, click, **Install**, click **Add** in the resulting Window, **select the RPM** in the third window, Click **Add**, and go back to the second Window to either do a full Install by clicking the **Install** button, **or** click **Upgrade**.
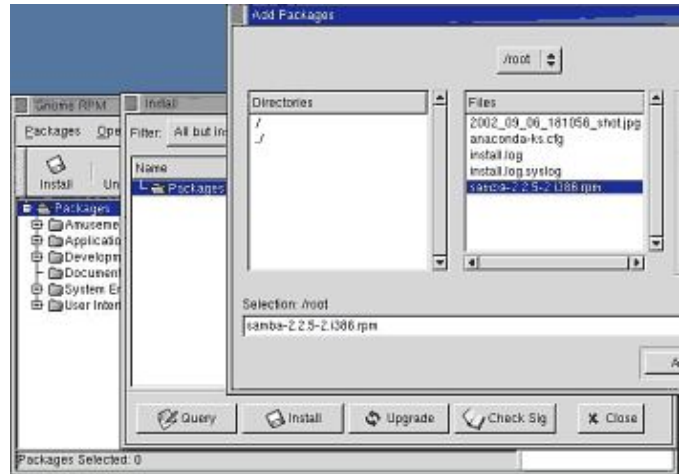
4. The Samba Web Administartion Tool (SWAT) is very useful for configuring Samba Users, modifying the Samba configuration file, and just about anything else that can be modified and maintained in Samba. SWAT is accessed with a web browser and consists of html pages. All of the Samba documentation is conveniently available from the SWAT home page. A Status display allows monitoring Samba and can even be used to shut–it–down or start–it–up manually. Samba does not have to be running to use SWAT.

The Samba RPM I used also installed SWAT. Some of them do not include SWAT. In those cases, SWAT is also available and can be installed as a separate RPM. The RH 7.3 distirbution does not include SWAT.

5. After Installing the RPM, you will find Samba and SWAT as separate entries in the RH Service Configuration. Samba should start if you have a valid smb.conf file, but SWAT may not. SWAT requires that either the xidentd (Extended Identification Protocol) or the identd (Identification Protocol) daemon is running and that a valid SWAT configuration file is present for xidentd or configuration lines are present in a couple of files for identd. The identd configuration lines are not present in the default RH or after the RPM installation. xidentd is newer, preferred, and after the RPM installation has a SWAT configuration file: **/etc/xident.d/swat**. However, there is a bug in the internal tcpwrappers functionality of xinetd and the supplied configuration file will not work without modification. Per the Red Hat "gotchas" (I don't personally like software with gotchas like these, and especially after they are still gotchas two years later––but, whacha want fur nut'n, anway?) one should use an editor such as qedit to change the following line:

    only_from = localhost

    to..

    only_from = 127.0.0.1

I went a step further. This is what my **/etc/xident.d/swat** file looks like:

```
# default: off
# description: SWAT is the Samba Web Admin Tool. Use swat \
# to configure your Samba server. To use SWAT, \
# connect to port 901 with your favorite web browser.
service swat
{
disable = no
port = 901
socket_type = stream
wait = no
# only_from = 127.0.0.1
user = root
server = /usr/sbin/swat
log_on_failure += USERID
}
```

I simply commented–out the offending line with a # and removed the restriction. This allows access to SWAT from any computer on our LAN by entering **http://linux:901/** in a browser **Address** box.

6. After saving the file make sure xidentd, Samba, and SWAT are started, the checkboxes are checked in Service Configuration, and the configuration is saved so they will start automatically during bootup.

There are many other possible Samba configurations. I leave you at this point to explore them now that you have SWAT as a tool with which to do so...

Larry

Please see our **CONTACT** page if you have any comments or corrections that would make this article better. Please use our **Forums** if you need help with a computer or network problem.