Samba Server

Step-by-Step Guide

May 8, 1999

By Ying Zhang
yzhang@sfu.ca

Contents

Revisions

May 8, 1999

- added section for RedHat 6.0 users
- added section on using smbmount

April 13, 1999

- some people had to chown the samba source directory after untarring it
- made some changes (Chapter 5)

March 15, 1999

- made some corrections (Chapter 10 ipchains)
- created a Postscript version for those that have problems with the PDF version

March 12, 1999

- updated the entire how-to, describes Samba 2.0.3
- renamed to step-by-step because it's not that "mini"
- created PDF version
- HTML version works with babelfish (but the examples will get garbled up)

October 22, 1998

- initial version of the mini-howto

Credits

Thanks to all the people who emailed me with their feedback, suggestions, and questions! I'm really glad my howtos have been able to help people out, please keep on visiting my site!

Thanks to the entire Samba development team for a job well done!

Updated Information

This section contains updated information that may supercede you see in the rest of the document. When I have more time, I will update this guide properly.

Redhat 6.0 Samba RPM

Redhat 6.0 comes with Samba 2.0.3 all packaged for you. If you are using Redhat 6.0, you can skip some parts of this howto (namely the parts on installing Samba). You may still want to follow the rest of the steps on creating the smbuser account, and you may want to use the example smb.conf file.

The RPM has smbfs built into it, which allows you to mount SMB shares from your Linux box. If you didn't use the RPM that came with Redhat 6.0 you can still compile smbfs manually, look at the configure script for more information. You can also take the .spec file from the samba SRPM package to see what they did.

Using SMBFS

The smbfs package isn't really a part of Samba (so I've read), but it comes with the newer Samba distributions. It is composed of two programs: **smbmount** and **smbumount**. You use these commands to mount and unmount remote SMB shares locally. In effect, from your Linux box you can use shares from a Windows box (or another box running Samba).

If you've ever used the earlier versions of smbfs, things have changed! Running smbmount by itself looks like running smbclient, so unless you look at the man pages for smbmount you won't know what's going on. Fortunately, the process isn't that difficult.

Let's say I have a share called **FOO** on my Windows 98 computer **BART** (\\bart\foo), and I want to mount this onto /mnt/foo on my Linux box **HOMER**. This is what I would type:

```
$ smbmount \\\\bart\\foo -c 'mount /mnt/foo'
```

Pretty simple isn't it? You need twice the number of \'s to backquote things. Make sure that the directory /mnt/foo already exists before running that command. To unmount this, I would type:

```
$ smbumount /mnt/foo
```

For more information, check out the man pages for smbmount, smbumount, and mount.

Introduction

This document will walk you through setting up Samba on your Linux box. At the end of this document, you will have:

- setup Samba on your Linux box with public shares
- setup Windows 95/98 clients to access the resources

This howto is **not a replacement** for the excellent documentation that comes with the Samba distribution! It is meant to help you get Samba up and running quickly, promise that you will read all the Samba documentation later.

Disclaimer

I cannot guarantee the accuracy or correctness of the information I present in this howto. I am in no way associated with the Samba team, what I describe here is basically a log of what I had to do to set up my Samba server. Do not blame me if something goes wrong, there are no warranties so use this information at your own risk!

On a brighter note, if you have any questions, comments, or suggestions, please feel free to send me email at yzhang@sfu.ca. I will try my best to answer your question or point you to better resources.

Background

My network at home consists of three computers, one Linux box and two Windows 98 boxes. My Linux box performs many tasks, one of which is serving files to my other computers via Samba.

For the sake of clarify, I will give these boxes some names. My Linux box is called **Homer**; the two Windows boxes are called **Bart** and **Lisa**, respectively. They will all belong to the **Simpsons** workgroup. Homer's IP address is 192.168.0.1, Bart's is 192.168.0.2, and Lisa's is 192.168.0.3.

I will be storing homework, bookmarks, and other personal documents on the share called **data**. This should be accessible from both win1 and win2 (mapped to the **M:** drive).

I also want a share for files that everyone can access; this will be on a share called **public**. This will be accessible from both win1 and win2 (mapped to the **P:** drive) and also by everyone with an account on homer.

Finally, homer will also be the primary WINS server for my network so that I don't have to maintain lmhosts files on Bart and Lisa.

Requirements

In writing this document, I will assume that you already have:

- a basic understanding of what Samba is and why you want it
- a working TCP/IP network
- a working Linux box on which you will install Samba
- the necessary packages to compile programs in Linux
- some working Windows 95/98 boxes waiting to access the shares
- you don't already have Samba running, if you do stop and remove it

Configuration Plan

Now you must plan your Samba configuration. I will tell you about mine; some of it may apply to you as well.

Homer is running Redhat Linux 5.1 with the 2.2.3 kernel, so I will be explaining how to install Samba the Redhat way (via RPM's). If you are using another distribution you probably won't be able to follow exactly what I do, so keep that in mind.

Bart and Lisa are both running Windows 98, not much should change if you Windows 95. If you are running Windows NT the steps will be a bit different, but the concepts are the same.

Since I only have three computers on the network and I'm not concerned about security, I will be using share level security in Samba. In fact, I use a single account on both Bart and Lisa and I use TweakUI to automatically login for me.

Getting Samba

As of this writing, the latest stable version of Samba is 2.0.3. You will need the source distribution, it is called **samba-2.0.3.tar.gz**, grab it from the Samba homepage (http://www.samba.org).

Installing Samba

First thing to do is extract the source distribution. Assuming you've downloaded it to /tmp, do this (you don't need to be root yet):

```
$ tar -zxvf samba-2.0.3.tar.gz
```

If you didn't get any error messages, you should have the extracted files in a directory called samba-2.0.3. The Samba source distribution comes with a bunch of packaging scripts. Of particular interest is the one that builds an RPM for us. Let's put blind-faith into this script and run it (now you have to be root):

```
$ su
# chown -R root:root samba-2.0.3
# cd samba-2.0.3/packaging/RedHat
# sh makerpms.sh
```

That should take a little bit of time while it builds the RPM. If that failed then something is wrong, uh-oh time to read the Samba docs. If it worked, these files will be created:

- /usr/src/redhat/RPMS/i386/samba-2.0.3-19990228.i386.rpm
- /usr/src/redhat/SRPMS/samba-2.0.3-19990228.src.rpm

Magical isn't it. Okay, let's install this sucker:

```
# rpm -Uvh /usr/src/redhat/RPMS/i386/samba-2.0.3-19990228.i386.rpm
```

Wow, can things get any easier? Next we will create a new user and group, then some directories. Then we modify a couple of files to use the Samba Web Administration Tool (SWAT).

Creating Stuff

I can't think of a better name for this section, but what we do here is create a bunch of user accounts, directories and files.

New User and Group

Since we are using share level security, we have to specify a guest account. This user account will be called **smbuser** and it will belong to the **smb** group.

All the files that Bart and Lisa write to the data and public shares will be owned by smbuser and belong to the smb group.

If you have Linuxconf, use it to do the dirty work. Otherwise, use groupadd and useradd to create the new accounts.

1. Create the group smb
2. Create the user smbuser, the home directory should be in /home/public

Disable login on the smbuser account since no one will actually be logging in with that name.

Public Directory

We should already have a /home/public (since you created it just a sec ago). It should be owned by smbuser and belong to the group smb. Since files in this directory should always be owned by the smb group, we will set the SGID bit on it. Everyone will have shared read and write access to this directory as well, so the permissions should be set like so:

```
# chown smbuser:smb /home/public
# chmod 2777 /home/public
```

Now every file that gets created in /home/public gets owned by the smb group no matter who creates it.

Data Directory

My data directory is /home/samba/data. This should only be accessible by certain people - those that belong to the smb group. So I set up the directory like so:

```
# mkdir /home/samba
# chown smbuser:smb /home/samba
# chmod 2770 /home/samba

# mkdir /home/samba/data
```

```
# chown smbuser:smb /home/samba/data
# chmod 2770 /home/samba/data
```

Again notice that I set the SGID bit on these directories so that anything created inside them get owned by the smb group.

Since I want access to this directory, I add myself to the smb group. Do this via Linuxconf or the usermod command. Or manually edit the /etc/group file if you feel brave :).

Check your hosts file

You should have an /etc/hosts file that maps host names to IP addresses. Mine would look like this:

```
# /etc/hosts

127.0.0.1     localhost    localhost.localdomain
192.168.0.1   homer
192.168.0.2   bart
192.168.0.3   lisa
```

Create an lmhosts file

The lmhosts file maps hostnames (or IP addresses) to NetBIOS (computer) names. Normally you would probably want the NetBIOS name and the hostname to be the same:

```
# /etc/lmhosts

localhost    homer
bart         bart
lisa         lisa
```

It looks a little silly doesn't it, but the thing on the left-hand side is the hostname (or IP address) and the thing on the right is the NetBIOS name.

Using SWAT

To enable SWAT, we have to check our /etc/services and /etc/inetd.conf files. By default, SWAT runs from port 901. You can change this if you like but for now let's leave it as is.

Let's check the /etc/services file for a line that looks like this:

```
swat            901/tcp             # Add swat service used via inetd
```

If it's there then you're okay. If you don't see a line like that, add it to the end of your services file. This is just a mapping that says "the service called swat runs on tcp port 901".

Next we check the /etc/inetd.conf file for a line that looks like this:

```
swat stream tcp nowait.400 root /usr/sbin/swat swat
```

Securing SWAT

If you don't care about securing SWAT then you can skip this section, but I feel much better having TCP wrappers protecting this guy, so I change the line to look like this:

```
swat   stream tcp nowait.400 root   /usr/sbin/tcpd /usr/sbin/swat
```

If you don't know what TCP wrappers are, read the man pages:

```
# man hosts.allow
```

Now your /etc/hosts.deny should have a single line in it that says ALL: ALL (it does right?). So we have to add this to your /etc/hosts.allow:

```
swat:           127.0.0.1   192.168.0.
```

This allows Homer, Bart, and Lisa to access SWAT.

Running SWAT

First let's restart inetd:

```
# killall -HUP inetd
```

Now with your favorite web browser go to the URL **http://192.168.0.1:901**. If all goes well you should see a box pop up asking for a username and password. For the username, enter in **root**, for the password enter your root password.

Configuring Samba

Okay SWAT is up and running, go click on everything to familiarize yourself with it. SWAT is a nice interface that creates the /etc/smb.conf file for you. If you have an existing smb.conf it will clobber the formatting and comments in it.

That's okay, because SWAT writes very nice and clean configuration files. Anyhow instead of walking you through each screen in SWAT, I'm going to cheat and just show you my /etc/smb.conf file instead. (The computer and workgroup names have been changed to protect my system identities).

Take a look at this and simply copy it overtop of your existing /etc/smb.conf. Make appropriate changes for your systems:

```
# Samba config file created using SWAT
# from bart (192.168.0.2)
# Date: 1999/02/08 18:27:26

# Global parameters
        workgroup = SIMPSONS
        server string = Samba SMB Server
        interfaces = 192.168.0.1/24 127.0.0.1/24
        bind interfaces only = Yes
        security = SHARE
        log file = /var/log/samba/log.%m
        max log size = 50
        read bmpx = No
        time server = Yes
        socket options = TCP_NODELAY
        os level = 65
        preferred master = Yes
        dns proxy = No
        wins support = Yes
        guest account = smbuser
        hide dot files = No

[public]
        comment = Public
        path = /home/public
        read only = No
        create mask = 0664
        directory mask = 0775
        guest ok = Yes

[data]
        comment = Data
        path = /home/samba/data
        read only = No
        create mask = 0660
        directory mask = 0770
        guest ok = Yes
```

Starting Samba

You can start Samba one of two ways: using SWAT or from the command line. It you want to use SWAT, go to the Status page and click start SMBD and NMBD. If it was already started, click stop and start to restart it to read your new configuration file.

To start from the command line:

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

Okay, everything is running and you didn't see error messages, right? Great! Let's give it a quick test.

Testing Samba

Okay, let's see if it's actually doing anything. Run:

```
# smbclient -L localhost
```

It might ask for a password, if so just press ENTER. You should now see stuff resembling:

```
Domain=[SIMPSONS] OS=[Unix] Server=[Samba 2.0.3]
        Sharename      Type      Comment
        ---------      ----      -------
        public         Disk      Public
        data           Disk      Data
        IPC$           IPC       IPC Service (Samba SMB Server)

        Server               Comment
        ---------            -------
        HOMER                Samba SMB Server

        Workgroup            Master
        ---------            -------
        SIMPSONS             HOMER
```

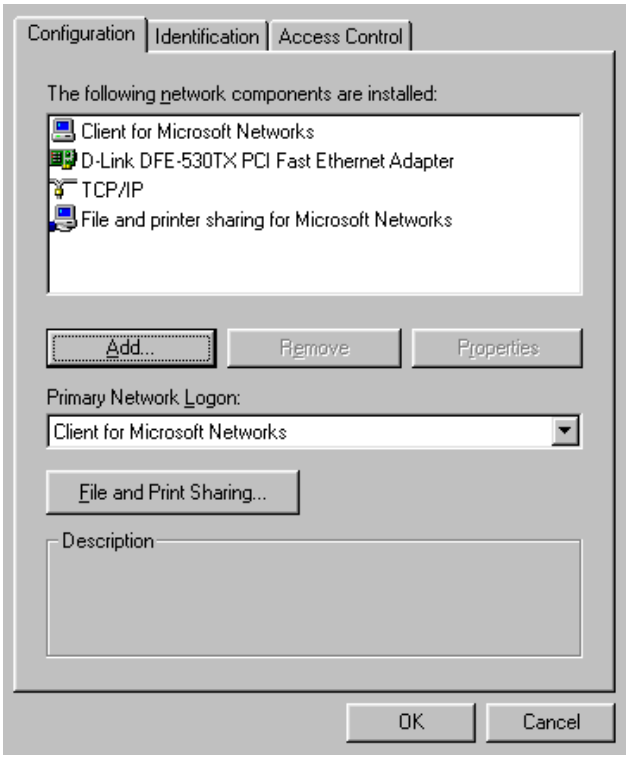
Everything's working? Great! Now all we have left is to configure your Windows boxes.

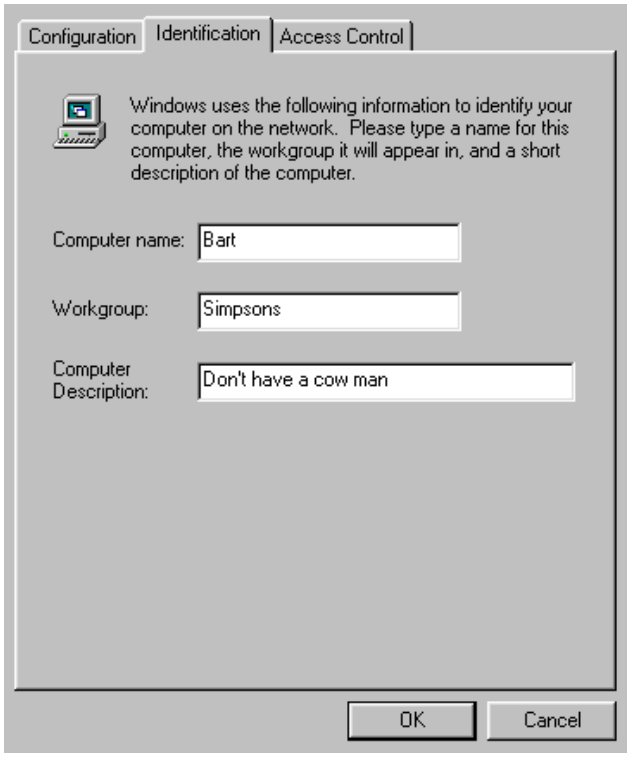
Configuring Windows

With the Samba server running on Homer, it's now time to turn our attention to Bart and Lisa. This part is really simple, just like connecting to any other Windows share. First, let's make sure the networking properties are set up properly.

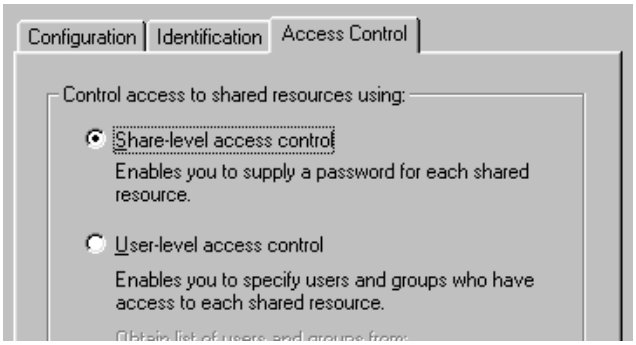
In the Identification tab of your Network setup, make sure your Computer Name and Workgroup is set up properly. Following our example so far, Bart should have the computer name Bart and the workgroup name Simpsons. The same goes for Lisa.

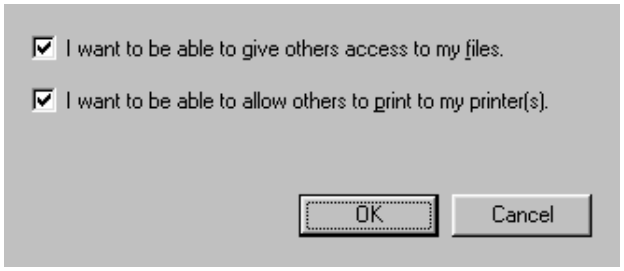
Here are some screenshots of my network properties:



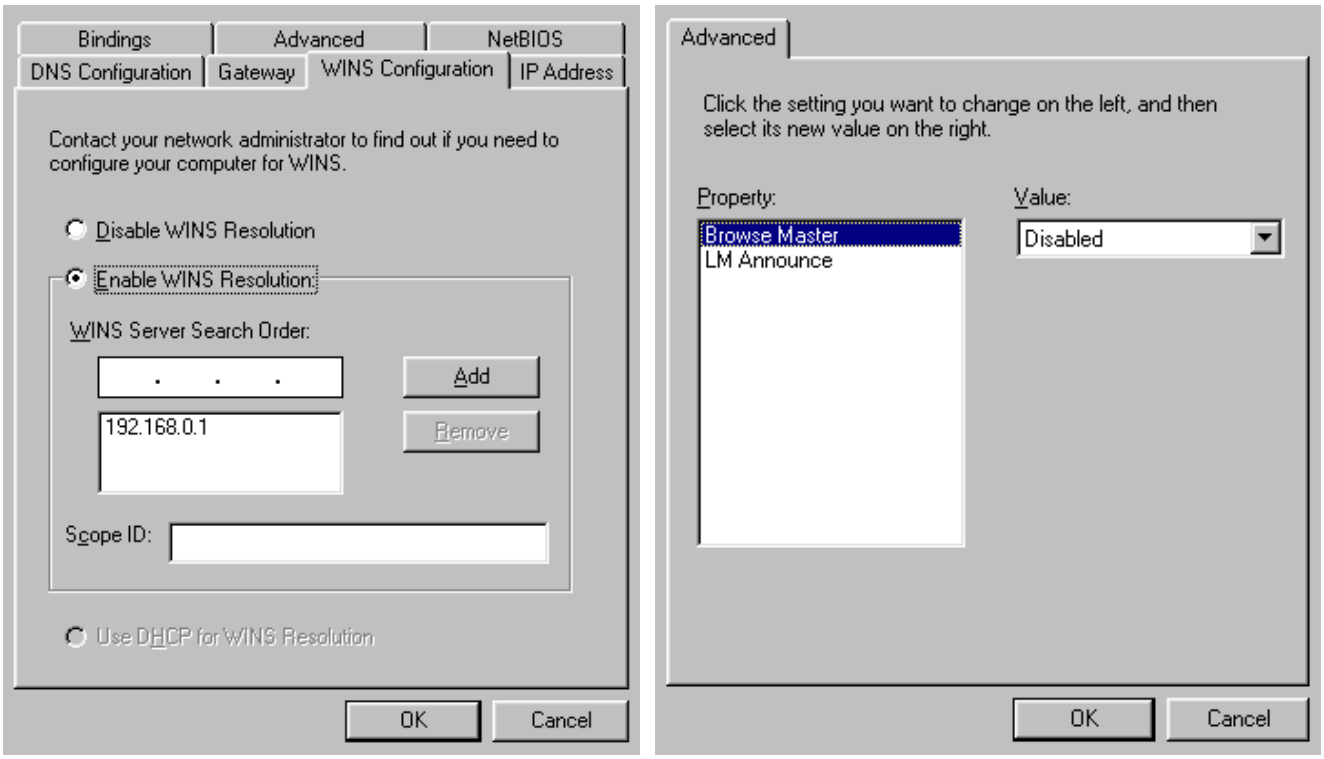


In the Access Control tab, enable Share-level access control. Next we look at the Configuration tab. We also need file and print sharing turned on.





Finally we tell Windows that Homer is the WINS server, and tell it not to act as the browse master (under File and Printer sharing).



In Windows NT, the configuration screens are different but the concept is the pretty much the same. A reader Donald Saltarelli points out that "NT clients with SP3 will need to enable clear-text passwords". If you are using NT, please check see the Samba documentation for more details.

You have all the information you need. Hit Apply and Ok and we should be all done! Windows will probably want you to reboot (is there any change that doesn't require a reboot?).

After Windows reboots, go to Network Neighborhood and you should see your Samba server. If you double click on it you should see the two shares, data and public. Now you can map them to drive letters if you so desire. Congratulations, everything is done!

Now we will look at securing our Samba server from outside attacks.

Securing Samba

Anytime you add a service to your machine you are giving crackers one more place to attack. Here we will look at some ways to protect your Samba server.

With Samba

In the Samba configuration (/etc/smb.conf), you can tell it which IP addresses to listen to. These lines are:

```
interfaces = 192.168.0.1/24 127.0.0.1/24
bind interfaces only = Yes
```

Of course you would substitute that for your own IP ranges. Because I'm paranoid, I add another layer of protection by filtering out the NetBIOS ports.

Filtering Ports

SMB uses ports 137-139, to be safe I block out both TCP and UDP ports 137-139. If you are using the 2.0 series kernel, you will be using a tool called ipfwadm to do this. With kernel 2.1 and 2.2, you use ipchains.

Using ipfwadm

Make sure you have ipfwadm, if not you can grab the RPM package from http://www.rpmfind.net/linux/RPM.

Add these lines to your /etc/rc.local file:

```
ipfwadm -I -P tcp -a deny -S any/0 137:139 -W eth0
ipfwadm -I -P udp -a deny -S any/0 137:139 -W eth0
ipfwadm -O -P tcp -a deny -S any/0 137:139 -W eth0
ipfwadm -O -P udp -a deny -S any/0 137:139 -W eth0
```

This will deny all incoming and outgoing TCP and UDP packets for ports 137-139 on interface eth0. eth0 is the NIC that connects my box to the Internet, you may have to modify these commands to suite your system configuration. Read the man ipfwadm for more information.

Using ipchains

You need the ipchains package for this, I think you can find it from http://www.rpmfind.net/linux/RPM.

Add these line to your /etc/rc.local

```
ipchains -A input -p tcp -j DENY --destination-port 137:139 -i eth0
ipchains -A input -p udp -j DENY --destination-port 137:139 -i eth0
ipchains -A output -p tcp -j DENY --destination-port 137:139 -i eth0
ipchains -A output -p udp -j DENY --destination-port 137:139 -i eth0
```

This does the same thing as the ipfwadm commands from the previous section.

Password Authentication

If you need user accounts and password authentication, you should investigate the other authentication methods in Samba (e.g. user-level security, domain level security, etc.). This is beyond the scope of this document.

Conclusion

This concludes this step-by-step howto. If you need more information about Samba, visit their homepage (http://www.samba.org) and read their documentation.

I hope you've found this document useful!