Configure the Samba Server on Linux

Introduction

Samba is a suite of utilities that allows your Linux box to share files and other resources, such as printers, with Windows boxes. This chapter describes how you can make your Linux box into a Windows Primary Domain Controller (PDC) or a server for a Windows Workgroup. Either configuration will allow everyone at home to have:

- their own logins on all the home windows boxes while having their files on the Linux box appear to be located on a new Windows drive
- shared access to printers on the Linux box
- Shared files accessible only to members of their Linux user group.

What's the difference between a PDC and Windows Workgroup member? A detailed description is beyond the scope of this chapter, but this simple explanation should be enough:

- A PDC stores the login information in a central database on its hard drive. This allows each user to have a universal username and password when logging in from all PCs on the network.
- In a Windows Workgroup, each PC stores the usernames and passwords locally so that they are unique for each PC.

This chapter will only cover the much more popular PDC methodology used at home. By default, Samba mimics a Windows PDC in almost every way needed for simple file sharing. Linux functionality doesn't disappear when you do this. Samba Domains and Linux share the same usernames so you can log into the Samba based Windows domain using your Linux password and immediately gain access to files in your Linux user's home directory. For added security you can make your Samba and Linux passwords different.

When it starts up, and with every client request, the Samba daemon reads the configuration file /etc/samba/smb.conf to determine its various modes of operation. You can create your own smb.conf using a text editor or the Web-based SWAT utility which is easier. Keep in mind, however, that if you create /etc/samba/smb.conf with a text editor then subsequently use SWAT to edit the file, you will lose all the comments you inserted with the text editor. I'll explain how to use both SWAT and a text editor to configure Samba later in this chapter.

Note: As your smb.conf is constantly being accessed, you're better off editing a copy of it if you decide not to use SWAT. After completing your modifications, test the validity of the changes using the testparm utility outlined in Chapter 12, "Samba Security and Troubleshooting", and when you are satisfied with your changes, copy the file back to its original location.

Download and Install Packages

Most RedHat and Fedora Linux software products are available in the RPM format. Downloading and installing RPMs isn't hard. If you need a refresher, Chapter 6, "Installing Linux Software", covers how to do this in detail.

Samba is comprised of a suite of RPMs that come on the Fedora CDs. The files are named:

- **samba**
- **samba-common**
- **samba-client**
- **samba-swat**

When searching for the file, remember that the RPM's filename usually starts with the RPM name followed by a version number as in samba-client-3.0.0-15.i386.

How to Get Samba Started

- You can configure Samba to start at boot time using the chkconfig command:

```
[root@bigboy tmp]# chkconfig smb on
```

- You can start/stop/restart Samba after boot time using the smb initialization script as in the examples below:

```
[root@bigboy tmp]# service smb start
[root@bigboy tmp]# service smb stop
[root@bigboy tmp]# service smb restart
```

Note: Unlike many Linux packages, Samba does not need to be restarted after changes have been made to its configuration file, as it is read after the receipt of every client request.

- You can test whether the smb process is running with the pgrep command, you should get a response of plain old process ID numbers:

```
[root@bigboy tmp]# pgrep smb
```

The Samba Configuration File

The /etc/samba/smb.conf file is the main configuration file you'll need to edit. It is split into five major sections, which Table 10-1 outlines:

Table 10-1: File Format - smb.conf

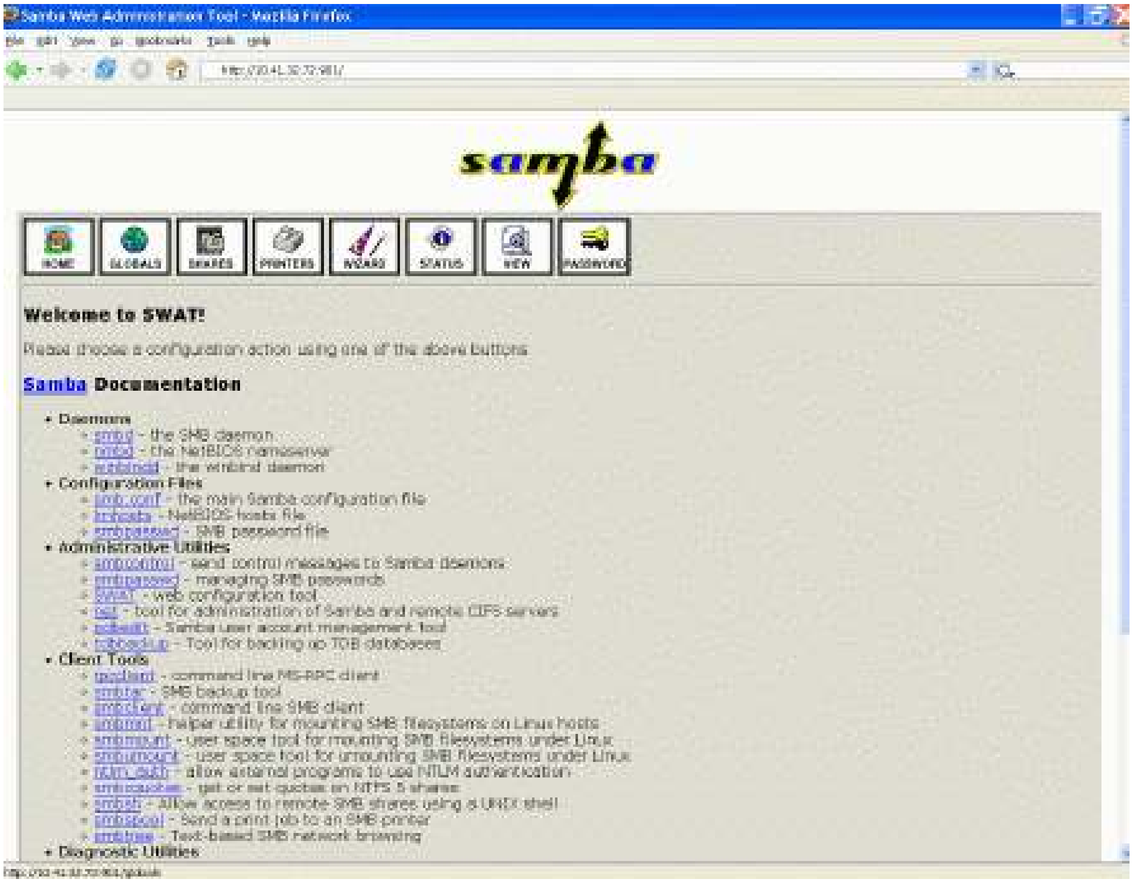
Section	Description
[global]	General Samba configuration parameters
[printers]	Used for configuring printers
[homes]	Defines treatment of user logins
[netlogon]	A share for storing logon scripts. (Not created by default.)
[profile]	A share for storing domain logon information such as "favorites" and desktop icons. (Not created by default.)

You can edit this file by hand, or more simply through Samba's SWAT web interface.

How SWAT Makes Samba Simpler

SWAT, Samba's web based configuration tool enables you configure your smb.conf file without you needing to remember all the formatting. Each SWAT screen is actually a form that covers a separate section of the smb.conf file into which you fill in the desired parameters. For ease of use, each parameter box has its own online help. Figure 10-1 shows the main SWAT login screen.

Figure 10-1 Samba SWAT Main Menu


Samba Web Administration Tool - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://10.41.32.72:901/
samba
HOME
GLOBALS
SHARES
PRINTERS
WIZARD
STATUS
VIEW
PASSWORD

Welcome to SWAT!

Please choose a configuration action using one of the above buttons

Samba Documentation

- Daemons
  - smbd - the SMB daemon
  - nmbd - the NetBIOS nameserver
  - winbindd - the winbind daemon
- Configuration Files
  - smb.conf - the main Samba configuration file
  - lmhosts - NetBIOS hosts file
  - smbpasswd - SMB password file
- Administrative Utilities
  - smbcontrol - send control messages to Samba daemons
  - smbpasswd - managing SMB passwords
  - SWAT - web configuration tool
  - net - tool for administration of Samba and remote CIFS servers
  - pdbedit - Samba user account management tool
  - tdbbackup - Tool for backing up TDB databases
- Client Tools
  - rpcclient - command line MS-RPC client
  - smbtar - SMB backup tool
  - smbclient - command line SMB client
  - smbmnt - helper utility for mounting SMB filesystems on Linux hosts
  - smbmount - user space tool for mounting SMB filesystems under Linux
  - smbumount - user space tool for unmounting SMB filesystems under Linux
  - ntlm_auth - allow external programs to use NTLM authentication
  - smbcquotas - get or set quotas on NTFS 5 shares
  - smbsh - Allow access to remote SMB shares using a UNIX shell
  - smbspool - Send a print job to an SMB printer
  - smbtree - Text-based SMB network browsing
- Diagnostic Utilities

http://10.41.32.72:901/globals

Basic SWAT Setup

You must always remember that SWAT edits the smb.conf file but also strips out any comments you may have manually entered into it beforehand. The original Samba smb.conf file has many worthwhile comments in it, you should save a copy as a reference before proceeding with SWAT. For example, you could save the original file with the name /etc/samba/smb.conf.original as in:

```
    [root@bigboy tmp]# cp /etc/samba/smb.conf
/etc/samba/smb.conf.original
```

As you can see, using SWAT requires some understanding of the smb.conf file parameters because it eliminates these comments. Become familiar with the most important options in this file before proceeding with SWAT.

SWAT doesn't encrypt your login password. Because this could be a security concern in a corporate environment you might want to create a Samba administrator user that has no root privileges or only enable SWAT access from the GUI console or localhost.

The enabling and disabling, starting and stopping of SWAT is controlled by xinetd, which is covered in Chapter 16, **"Telnet, TFTP, and xinetd",** via a configuration file named /etc/xinetd.d/swat. Here is a sample:

```
service swat
{

      port             = 901
      socket_type      = stream
      protocol         = tcp
      wait             = no
      user             = root
      server           = /usr/sbin/swat
      log_on_failure   += USERID
      disable          = no
      only_from        = localhost

}
```

The file's formatting is fairly easy to understand, especially as there are only two entries of interest.

- The disable parameter must be set to no to accept connections. This can automatically be switched between yes and no as we will see later.

- The default configuration only allows SWAT web access from the VGA console only as user root on port 901 with the Linux root password. This means you'll have to enter "**http://127.0.0.1:901**" in your browser to get the login screen.

You can make SWAT accessible from other servers by adding IP address entries to the only_from parameter of the SWAT configuration file. Here's an example of an entry to allow connections only from 192.168.1.3 and localhost. Notice that there are no commas between the entries.

```
only_from = localhost 192.168.1.3
```

Therefore in this case you can also configure Samba on your Linux server bigboy IP with address 192.168.1.100 from PC 192.168.1.3 using the URL http://192.168.1.100:901.

Remember that most firewalls don't allow TCP port 901 through their filters. You may have to adjust your rules for this traffic to pass.

Controlling SWAT

As with all xinetd-controlled applications, the chkconfig command automatically modifies the disable field accordingly in the configuration file and activates the change.

Before SWAT can be used, the xinetd program which controls it must be activated in advance.

You can start/stop/restart xinetd after boot time using the xinetd initialization script as in the examples below:

```
[root@bigboy tmp]# service xinetd start
[root@bigboy tmp]# service xinetd stop
[root@bigboy tmp]# service xinetd restart
```

Just like most Linux systems applications, you can configure xinetd to start at boot time using the chkconfig command:

```
[root@bigboy tmp]# chkconfig xinetd on
```

To activate SWAT use:

```
        [root@bigboy tmp] chkconfig swat on
```

To deactivate SWAT use:

```
        [root@bigboy tmp] chkconfig swat off
```

Encrypting SWAT

By default SWAT is configured via an unencrypted web link using the Linux root account. When running SWAT in the unsecured mode above you should take the added precaution of using it from the Linux console whenever possible.

You can configure SWAT to work only with securely encrypted HTTP (HTTPS) versus the regular HTTP method shown above. Here is how it's done. (Please refer to the VPN section of Appendix I, "Miscellaneous Linux Topics," for more details on encryption methods.)

Create an stunnel User

You can create a stunnel user via the useradd command:

```
        [root@smallfry tmp]# useradd stunnel
```

Create the Certificates

From the /usr/share/ssl/certs directory and create the encryption key certificate using the make command. Use all the defaults when prompted, but make sure you use the server's IP address when prompted for your server's Common Name or hostname.

```
        [root@bigboy tmp]# cd /usr/share/ssl/certs
        [root@bigboy certs]# make stunnel.pem
        ...
        Common Name (eg, your name or your server's hostname) []:
172.16.1.200
        ...
        [root@bigboy certs]#
```

Note: The resulting certificate has only a 365 day lifetime. Remember to repeat this process next year.

Modify Certificate File Permissions

The certificate needs to only be read by root and the stunnel user. Use the chmod and chgrp commands to do this.

```
        [root@bigboy certs]# chmod 640 stunnel.pem
        [root@bigboy certs]# chgrp stunnel stunnel.pem

        [root@bigboy certs]# ll /usr/share/ssl/certs
        -rw-r-----  1 root stunnel   1991 Jul 31 21:50 stunnel.pem
        [root@bigboy certs]#
```

Create an /etc/stunnel/stunnel.conf Configuration File

You can configure the stunnel application to:

- Intercept encrypted SSL traffic received on any TCP port
- Decrypt this traffic

- Funnel the unencrypted data to any application listening on another port.

For example, you can configure the /etc/stunnel/stunnel.conf file to intercept SSL traffic on the SWAT port 901 and funnel it decrypted to a SWAT daemon running on port 902. Here's how:

```
# Configure stunnel to run as user "stunnel" placing temporary
# files in the /home/stunnel/ directory
chroot  = /home/stunnel/
pid     = /stunnel.pid
setuid  = stunnel
setgid  = stunnel

# Log all stunnel messages to /var/log/messages
debug   = 7
output  = /var/log/messages

# Define where the SSL certificates can be found.
client  = no
cert    = /usr/share/ssl/certs/stunnel.pem
key     = /usr/share/ssl/certs/stunnel.pem

# Accept SSL connections on port 901 and funnel it to
# port 902 for swat.
[swat]
accept   = 901
connect  = 902
```

Create A New /etc/xinetd.d File For Secure SWAT

To start, copy the swat file and name it swat-stunnel. We then configure the new file to be enabled, listening on port 902 and accepting connections only from localhost. We also make sure that the service is set to swat-stunnel.

```
[root@bigboy certs]# cd /etc/xinetd.d
[root@bigboy xinetd.d]# cp swat swat-stunnel
```

Your new swat-stunnel file should look like this:

```
service swat-stunnel
{
       port            = 902
       socket_type     = stream
       wait            = no
       only_from       = 127.0.0.1
       user            = root
       server          = /usr/sbin/swat
       log_on_failure  += USERID
       disable         = no
       bind            = 127.0.0.1
}
```

Disable SWAT in the /etc/xinetd.d/swat File

The stunnel daemon actually intercepts port 901 traffic on behalf of swat-stunnel. You'll need to disable SWAT to prevent a conflict.

Edit the /etc/services file To create a Secure SWAT entry

The xinetd daemon searches /etc/services file for ports and services that match those listed in each configuration file in the /etc/xinetd.d directory. If the daemon doesn't find a match it ignores the configuration file.

We now have to edit /etc/services to include our new swat-stunnel file like this.

```
      swat-stunnel    902/tcp     # Samba Web Administration Tool
(Stunnel)
```

Activate swat-stunnel

You can then start the new swat-stunnel application with the chkconfig command. You'll also need to shutdown regular swat beforehand.

```
[root@bigboy xinetd.d]# chkconfig swat off
[root@bigboy xinetd.d]# chkconfig swat-stunnel on
```

Start stunnel

Now start stunnel for the encryption to take place.

```
[root@bigboy xinetd.d]# stunnel
```

In Fedora Core 2 you may get a cryptonet error when starting stunnel as in:

```
Unable to open "/dev/cryptonet"
```

This is caused by an incompatibility with the hwcrypto RPM used for hardware-, not software-based encryption. You need to uninstall hwcrypto to get stunnel to work correctly.

```
[root@bigboy xinetd.d]# rpm -e hwcrypto
```

You will then have to stop stunnel, restart xinetd and start stunnel again. After this, stunnel should begin to function correctly. Unfortunately stunnel doesn't have a startup script in the /etc/init.d directory and needs to be terminated manually using the pkill command.

```
[root@bigboy xinetd.d]# pkill stunnel
[root@bigboy xinetd.d]# stunnel
```

Test Secure SWAT

Your Samba server should now be listening on both port 901 and 902 as shown by the netstat -an command that follows. The server will accept remote connections on port 901 only.

```
[root@bigboy xinetd.d]# netstat -an
...
...
tcp        0      0 0.0.0.0:901       0.0.0.0:*         LISTEN
tcp        0      0 127.0.0.:902      0.0.0.0:*         LISTEN
...
...
[root@bigboy xinetd.d]#
```

Test the Secure SWAT Login

Point your browser to the Samba server to make an HTTPS connection on port 901.
https://server-ip-address:901/
You will be prompted for the Linux root user username and password. There will be a delay of about 60 to 75 seconds with each login.

Troubleshooting Secure SWAT

Sometimes you'll make mistakes in the stunnel.conf file but changes to this file take effect only after stunnel has been restarted. Unfortunately, there is no stunnel script in the /etc/init.d directory to easily stop and restart it. You have to use the pkill command to stop it and the stunnel command to start it again:

```
[root@bigboy tmp]# pkill stunnel ; stunnel
```

Make sure the file permissions and ownership on the stunnel.pem file are correct and that SWAT is always permanently off, but swat-stunnel is permanently on.

You can also refer to Chapter 4, "Simple Network Troubleshooting", to isolate connectivity issues between the SWAT client and Samba server on TCP port 901 amongst other things.

How to Make SWAT Changes Immediate

SWAT immediately changes the functioning of Samba whenever you commit your changes through the web GUI.

Creating A Starter Configuration

I'll now illustrate how to configure a Samba server to be the PDC for a small network is by using SWAT. You'll need to edit the various sections of the smb.conf file, so I'll walk you through what you'll find in each.

The [Global] Section

The [global] section governs the general Samba settings. Table 10-2 explains the parameters you need to set in order to create a PDC.

Table 10-2: smb.conf Minimum Settings, "Global" Section

Parameter	Value	Description
domain logons	Yes	Tells Samba to become the PDC
preferred master	Yes	Makes the PDC act as the central store for the names of all windows clients, servers and printers on the network. Very helpful when you need to "browse" your local network for resources. Also known as a local master browser.
domain master	Yes	Tells Samba to become the master browser across multiple networks all over the domain. The local master browsers register themselves with the domain master to learn about resources on other networks.
os level	65	Sets the priority the Samba server should use when negotiating to become the PDC with other Windows servers. A value of 65 will usually make the Samba server win.
wins support	Yes	Allows the Samba server to provide name services for the network. In other words keeps track of the IP addresses of all the domain's servers and clients.
time server	Yes	Lets the samba server provide time updates for the domain's clients.
workgroup	"homenet"	The name of the Windows domain we'll create. The name you select is your choice. I've decided to use "homenet".
security	user	Make domain logins query the Samba password database located on the samba server itself.

Here's how to set the values using SWAT.

1. Log into SWAT and click on the [global] section.
2. Click the Advanced button to see all the options.
3. Make your changes and click on the Commit Changes button when finished.
4. Your smb.conf file should resemble the example below when you're finished. You can view the contents of the configuration file by logging in to the samba server via a command prompt and using the cat /etc/samba/smb.conf to verify your changes as you do them.

```
[global]
        workgroup = HOMENET
        time server = Yes
        domain logons = Yes
        os level = 65
        preferred master = Yes
        domain master = Yes
```

Note: security = user and WINS support = yes are default settings for Samba and they may not show up in your smb.conf file, even though you may see them in SWAT.

Using the SWAT Wizard

The SWAT utility has a Wizard button that can be used to configure your server as a PDC quickly. However the defaults may not be to your liking, for example, the default domain is MYGROUP and some of the [global] parameters mentioned previously will be set to auto.

The [homes] Section

Part of the process of adding a user to a Samba domain requires you to create a Linux user on the Samba PDC itself. When you log into the Samba PDC, you'll see a new drive, usually named Z:, added to your PC. This is actually a virtual drive that maps to the corresponding Linux users' login directories on the Linux PDC.

Samba considers all directories to be shares that can be configured with varying degrees of security. The [homes] section governs how Samba handles default login directories.

Table 10-3 explains the minimum settings you need to create a functional [Homes] section.

Table 10-3: smb.conf Minimum Settings, "Home" Section

Parameter	Value	Description
browseable	No	Doesn't allow others to browse the contents of the directory
read only	No	Allows the samba user to also write to their Samba Linux directory
create mask	0664	Makes new files created by the user to have "644" permissions. You want to change this to "0600" so that only the login user has access to files.
directory mask	0775	Makes new sub-directories created by the user to have "775" permissions. You want to change this to "0700" so that only the login user has access to directories.

Here's how to set the values using SWAT:

1. Click on the SWAT shares button to proceed to where shared directories are configured.
2. Click the Advanced button to see all the options.
3. Choose the Homes share.
4. Make your changes and click on the Commit Changes button when finished.

5. Your smb.conf file should resemble this when finished. You can view the contents of the configuration file by logging in to the samba server via a command prompt and using the cat /etc/samba/smb.conf to verify your changes as you do them.

```
[homes]
  read only = No
  browseable = No
  create mask = 0644
  directory mask = 0755
```

The [netlogon] and [profiles] Share Sections

The [netlogon] share section contains scripts that the windows clients may use when they log into the domain. The [profiles] share section stores settings related to the look and feel of windows so that the user has the same settings no matter which Windows PC is logged into. The [profiles] share section stores things such as favorites and desktop icons.

Your smb.conf file should look like this when you're finished:

```
[netlogon]
       path = /home/samba/netlogon
       guest ok = Yes

[profiles]
       path = /home/samba/profiles
       read only = No
       create mask = 0600
       directory mask = 0700
```

Here's how to do it.

1. Click the Shares button.
2. Create a [netlogon] share.
3. Modify the path and guest ok settings.
4. Click on the Commit Changes button.
5. Create a [profiles] share section.
6. Modify the path, mask and read only settings. The mask settings allow only the owner of the netlogon subdirectory to be able to modify its contents.
7. Click on the Commit Changes button.

Remember to create these share directories from the command line afterwards.

```
[root@bigboy tmp]# mkdir -p /home/samba/netlogon
[root@bigboy tmp]# mkdir -p /home/samba/profile
[root@bigboy tmp]# chmod -R 0755 /home/samba
```

The [printers] Share Section

Samba has special shares just for printers, and these are configured in the [printers] section of SWAT. There is also a share under [printers] called printers which governs common printer settings. Print shares always have the printable parameter set to yes. The default smb.conf [printers] share section looks like this:

```
[printers]
    comment = All Printers
    path = /var/spool/samba
    printable = Yes
    browseable = No
```

Shares for Specific Groups of Users

The default Samba Version 3 smb.conf file you saved at the beginning of this exercise has many varied examples that you may use and apply to your particular environment.

Samba Passwords

You should be aware that your Linux password and Samba passwords are stored in two different locations. This provides the Samba administer the flexibility of allowing only some of the Linux users to have Samba accounts.

Use the passwd command to change Linux passwords, which are stored in the /etc/shadow file. Samba passwords are stored in the /etc/samba/smbpasswd file and can be changed smbpasswd command.
This difference is important, as you will see throughout the chapter.

How to Create a Samba PDC Administrator User

To do both SWAT and user administration with Samba you'll need to create administrator accounts on the Samba PDC Linux server.

Home Environment

By default, the root user is the Samba administrator, and SWAT requires you to use the Linux root password to be used. Fortunately, you can add workstations to the Windows domain by creating a Samba specific root password. This is done using the smbpasswd command.

```
[root@bigboy tmp]# /usr/bin/smbpasswd -a root password
```

Note: Remember that regular Linux logins via the console, Telnet or SSH require the Linux passwd command. Samba domain logins use the smbpasswd password. Samba passwords are stored in the /etc/samba/smbpasswd file.

Corporate Environment

In a corporate environment, you may want more than one person to administer Samba, each with their own usernames. Here are the steps to do this

1. Create a Linux user group, such as sysadmin with the groupadd command.
2. Use SWAT to update your smb.conf file so that the sysadmin group is listed in the [global] parameter settings.

   ```
   domain admin group = @sysadmin
   admin users = @sysadmin
   printer admin = @sysadmin
   ```

3. Create individual Linux users that are part of this group.

4. Use the smbpasswd command to create Samba passwords for Domain logins for this group. For security reasons this password may be different from the Linux password used to log into the Linux system from the console, via telnet or ssh. (Remember that Linux passwords are changed with the passwd command.)

How To Add Workstations To Your Samba Domain

Adding workstations to a Samba domain is a two step process involving the creation of workstation trust accounts on the Samba server and then logging into each workstation to add them to the domain.

Create Samba Trust Accounts For Each Workstation

PDCs will accept user logins only from trusted PCs that have been placed in its PC client database. Samba can create these Machine Trusts in two ways, either manually or automatically.

Manual Creation Of Machine Trust Accounts (NT Only)

The commands in this example create a special Linux group for Samba clients and then add a special machine user that's a member of the group. The password for this user is then disabled and the machine is then added to the smbpasswd file to help keep track of which devices are members of the domain. In summary, a machine trust account needs to have entries in the /etc/passwd and /etc/smbpasswd files. Pay careful attention to the dollar sign ($) at the end and replace machine_name with the name of the Windows client machine.

```
[root@bigboy tmp]# groupadd samba-clients
[root@bigboy tmp]# /usr/sbin/useradd -g samba-clients \
-d /dev/null -s /bin/false machine_name$
[root@bigboy tmp]# passwd -l machine_name$
[root@bigboy tmp]# smbpasswd -a -m machine_name
```

This is the only way to configure machine trusts using Windows NT.

Dynamic Creation of Machine Trust Accounts

Although you can use the manual method, the recommended way of creating machine trust accounts is simply to allow the Samba server to create them as needed when the Windows clients join the domain which known as making a machine account on the fly. You can set this up by editing the /etc/samba/smb.conf file to automatically add the required users.

The easiest way to do this using SWAT in the Global menu to modify the add machine script parameter.

```
[global]
# <...remainder of parameters...>
add machine script = /usr/sbin/useradd -d /dev/null -g samba-
clients -s /bin/false -M %u
```

When you have completed the modifications, you'll need to create the samba-clients Linux group that will be used to help identify the all the domain's Windows clients listed in the /etc/passwd file.

```
[root@bigboy tmp]# groupadd samba-clients
```

In Samba version 2, you need to add the client to the smbpasswd file also

```
[root@bigboy tmp]# smbpasswd -a -m machine_name
```

Samba version 3 adds it automatically.

Make Your PC Clients Aware Of Your Samba PDC

There are many types of Windows installed on people's PCs and each version has its own procedure for joining a domain. The next sections show you how to add the most popular versions of Windows clients to your domain:

Windows 95/98/ME and Windows XP Home

Windows 9x machines do not implement full domain membership and therefore don't require machine trust accounts. Here's what you need to do:

1. Navigate to the Network section of the Control Panel (Start ->Settings->Control Panel->Network)
2. Select the Configuration tab
3. Highlight "Client for Microsoft Networks"
4. Click the Properties button.
5. Check "Log onto Windows NT Domain", and enter the domain name.
6. Click all the OK buttons and reboot!

Windows NT

For Windows NT, you must first create a manual Samba machine trust account as explained earlier, and then follow these steps:

1. Navigate to the Network section of the Control Panel (Start ->Settings->Control Panel->Network)
2. Select the "Identification" tab
3. Click the "Change" button
4. Enter the domain name and computer name, do not check the box Create a Computer Account in the Domain. In this case, the existing machine trust account is used to join the machine to the domain.
5. Click "OK". You should get "Welcome to <DOMAIN>" message as confirmation that you've been added.
6. Reboot.

You can now log in using any account in the /etc/smbpasswd file with your domain as the domain name.

Windows 200x and Windows XP Professional

For the 200x and XP Professional varieties of Windows, create a dynamic Samba machine trust account, then go through these steps:

1. Press the Windows and Break keys simultaneously to access the System Properties dialogue box.
2. Click on the 'Network Identification' or 'Computer Name' tab on the top.
3. Click the "Properties" button.
4. Click on the "Member of Domain" button.
5. Also enter your domain name and computer name and then click "OK"
6. You will be prompted for a user account and password with rights to join a machine to the domain. Enter the information for your Samba administrator. In this home environment scenario, the user would be root with the corresponding smbpasswd password. Now, you should get a "Welcome to <DOMAIN>" message confirming that you've been added.
7. Reboot.

Log in using any account in the /etc/smbpasswd file with your domain as the domain name.

Note: With Samba version 2 you may also have to make a few changes to your system's registry using the regedit command and reboot before continuing.

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Netlogon\Parameter
s]
"requiresignorseal"=dword:00000000
"signsecurechannel"=dword:00000000
```

How To Add Users To Your Samba Domain

Adding users to a domain has three broad phases. The first is adding a Linux user on the Samba server, the second is creating a Samba smbpasswd that maps to the new Linux user created previously, and the third is to map a Windows drive letter to the user's Linux home directory. Let's take a closer look:

Adding the Users in Linux

First, go through the process of adding users in Linux just as you would normally. Passwords won't be necessary unless you want the users to log in to the Samba server via telnet or ssh.

Create the user

To create the user, use the command:

```
[root@bigboy tmp]# Useradd -g 100 peter
```

Give them a Linux Password

Giving them a Linux password is only necessary if the user needs to log into the Samba server directly. If the user does, use this method:

```
[root@bigboy tmp]# passwd peter
Changing password for user peter.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@bigboy tmp]#
```

Mapping The Linux Users To An smbpassword

Next, you need to create Samba domain login passwords for the user

```
[root@bigboy tmp]# /usr/bin/smbpasswd -a username password
```

The -a switch adds the user to the /etc/smbpasswd file. Use a generic password then have users change it immediately from their workstations in the usual way.
Remember the smbpasswd sets the Windows Domain login password for a user, which is different from the Linux login password to log into the Samba box.

Mapping A Private Windows Drive Share

By default, Samba automatically gives each user logged into the domain an H: drive that maps to the /home/username directory on the Linux box.

Mapping Using "My Computer"

If the auto-mapping doesn't work then try:

1. Let the user log into the domain.
2. Right-click on the "My Computer" icon on the desktop.
3. Click on "Map Network Drive".
4. Select a drive letter.
5. Browse to the HOMENET domain, then the Samba server, then the user's home directory.
6. Click on the check box "Reconnect at Logon", to make the change permanent

Mapping from the Command Line

If you find the "My Computer" method too time consuming for dozens of users or if the PC doesn't have the feature available, then you can use the command-line method and possibly make it into a script.

1. Create a master logon batch file for all users

   ```
   [root@bigboy tmp]# vi /home/samba/netlogon/login.bat
   ```

2. Add the following lines to mount the user's share as drive P: (for private).

   ```
   REM Drive Mapping Script
   net use P: \\bigboy\
   ```

3. Make the file world readable using:

   ```
   [root@bigboy tmp]# chmod 644 /home/samba/netlogon/login.bat
   ```

4. Linux and Windows format text files slightly differ. As the file resides on a Linux box, but will be interpreted by a Windows machine, you'll have to convert the file to the Windows format. Use the unix2dos command.

   ```
   [root@bigboy tmp]# unix2dos /home/samba/netlogon/login.bat
   unix2dos: converting file /home/samba/netlogon/login.bat
   to DOS format ...
   [root@bigboy tmp]#
   ```

5. The final step is to edit your smb.conf file's [global] section have a valid entry for the logon script parameter.

This can be done using SWAT via the Globals menu.

```
[global]
  logon script = login.bat
```

Now your users will have additional disk space available on a Windows P: drive whenever they login.

Domain Groups And Samba

Samba supports domain groups that will allow users who are members of the group to be able to have Administrator rights on each PC in the domain. This enables them to add software and configure network settings. In Windows, Domain Groups also have the ability to join machines to the domain: however, Samba does not support this currently.

The domain admin group parameter specifies users who will have domain administrator rights. The argument is a space-separated list of user names or group names (group names must have an @ sign prefixed). For example:

```
domain admin group = USER1 USER2 @GROUP
```

How To Delete Users From Your Samba Domain

Deleting users from your Samba domain is a two stage process in which you have to remove the user from the Linux server and also remove the user's corresponding smbpasswd entry. Here's how:

1. Delete the users using the smbpasswd with the -x switch

   ```
    [root@bigboy tmp]# smbpasswd -x john
   Deleted user john.
    [root@bigboy root]#
   ```

2. Delete The Linux User by following the normal deletion process. For example, to delete the user john and all john's files from the Linux server use:

   ```
   [root@bigboy tmp]# userdel -r john
   ```

Sometimes you may not want to delete the user's files so that they can be accessed by other users at some other time. In this case you can just deactivate the user's account using the passwd -l username command.

How To Modify Samba Passwords

You can set your Samba server to allow users to make changes in their domain passwords and have these mirrored automatically in their Linux login passwords. Table 10-4 explains the [global] smb.conf parameters that you need to change.

Table 10-4: smb.conf Settings, Enabling Online Password Changes

Parameter	Value	Description
unix passwd sync	Yes	Enables Samba/Linux password synchronization
passwd program	Use the SWAT defaults	Lists the location of the Linux password file which is usually /bin/passwd.
passwd chat	Use the SWAT defaults	A short script to change the Linux password using the Samba password

By

Nazeer Ahmed (IT Support)