SAMBA

- **Server Message Block** (**SMB**) is the standard protocol that Windows uses for sharing files and print services, as well as communications abstractions such as named pipes and mail slots between computers.

- In order for Windows workstation to access UNIX shared files it must either use NFS (difficult since it is NFS challenged) or have the UNIX server use an application that will translate SMB into the protocol the NFS understands.

- A UNIX server running **SAMBA** can enable Windows-based computers to access the shares it is exporting under NFS.

- Samba is a file and print server for Windows-based clients using TCP/IP as the underlying transport protocol. It can support any SMB/CIFS-enabled client.

- SAMBA is an open source implementation of the SMB protocol used in Windows networking.

- It can be used to provide Windows users with "network neighborhood" access to Linux file systems and printers.

- ***smbd*** is the daemon that provides SMB file sharing and print services.

- ***nmbd*** is the daemon that provides name resolution for NetBIOS clients.

- ***/etc/samba/smb.conf*** is Samba's configuration file. It defines global options such as naming conventions, access permissions, logfiles and authentication rules.

- It also defines filesystem shares and the access permissions granted to users.

- ***smbclient*** lists shares on a running Samba server.

SMB Methodology

- Microsoft implements their own form of the SMB Protocol, to provide file and printer sharing in all versions of Windows.

- Traditionally, SMB uses the following port assignments and services:

Protocol	Port	Service Name	Function
UDP	135	Location Service loc-srv	RPC endpoint mapping
TCP	135	Location Service loc-srv	RPC endpoint mapping
UDP	137	NetBIOS Name Service netbios-ns	Translate NetBIOS names to IP addresses, much like DNS
TCP	138	NetBIOS Datagram Service netbios-dgm	
TCP	139	NetBIOS Session Service netbios-ssn	File & Printer data transfer
UDP	445	Direct Hosted Service microsoft-ds	Method of File & Printer data transfer

- In general, SMB sessions are established in the following order:

    - "TCP Connection" - establish 3-way handshake (connection) to port 139/tcp or 445/tcp.

    - "NetBIOS Session Request" - using the following "Calling Names": The local machine's NetBIOS name plus the 16th character 0x00; The server's NetBIOS name plus the 16th character 0x20

        - "SMB Negotiate Protocol" - determine the protocol dialect to use (Lanman (LAN Manager) for example).

    - SMB Session Startup. Passwords are encrypted (or not) according to one of the following methods:
        - Null (no encryption); Cleartext (no encryption)
        - LM and NTLM; NTLM; NTLMv2. The password is then hashed and sent to the computer requesting the session (ironically, *before* prompting for a password).

    - SMB Tree Connect: Connect to a share name (e.g., \\servername\share); Connect to a service type (e.g., IPC$ named pipe)

Installation

- Install Samba in the usual way:

  dnf install samba

Configuring SAMBA

- Before you can use Samba as a Windows file service provider, you have to configure it properly.

- When you configure Samba, you set its operational parameters, create entries for the directories that you want to share, and make network printers available.

- Samba configuration also allows you to establish user access permissions for shares. Virtually all parts of the Samba configuration process are managed by a central configuration file: **/etc/samba/smb.conf**.

- The syntax of the **smb.conf** file is pretty straightforward. The file is broken into sections, with the section names in square brackets.

- Within each section, parameters are set by statements in a "name = value" format. The easiest way to get familiar with the Samba configuration file is to work through the details of one. The following listing shows a sample *smb.conf file:*

Global Section

- The **[global]** section is used to configure parameters that apply to the server as a whole, and to provide defaults for other sections.

- For example:

```
[global]
        workgroup = CST323
        server string = Samba Server
        hosts allow = 192.168.1. 192.168.2. 127.
        printing = bsd
        printcap name = /etc/printcap
        load printers = yes
        #guest account = nfstest
        log file = /var/log/samba/log.%m
        max log size = 50
        security = share
        …….
```

- The first entry sets the Windows-Domain-Name or Workgroup-Name.

- The next entry server string is the equivalent of the Windows Description field

- The allow option is important for security. It allows you to restrict connections to machines that are on your local network.

- The example shown restricts access to two C class networks and the "loopback" interface.

- The next entry tells Samba what type of printing subsystem is available on your UNIX system. Different versions of UNIX handle printing differently.

- Following that you can automatically load your printer list rather than setting them up individually.

- Uncomment the next line only if you want a guest account, you must add this to ***/etc/passwd*** otherwise the user "nobody" is used

- This next option sets a separate log file for each client machine that connects.

- The next field sets the security mode. This will be discussed later in this section.

Homes Section

- The [homes] section allows clients to connect to a user's home directory, without having a specific entry for the directory in the ***smb.conf*** file.

- When a service request is made, the rest of the ***smb.conf*** file is searched to find the specific service that was requested.

- If the service is not found, and the [homes] section is present, the password file is searched to find the home directory for the user.

- Samba then makes the user's home directory available as a share by cloning the [homes] section entry. Here is a sample [homes] entry:

```
[homes]
  comment = Home Directories
  browseable = no
  writable = yes
```

- The "comment" field is used by a client attempting to see what shares are available on this server.

- The next parameter controls whether Samba will display home directories in the network browse list.

- The "writable" parameter controls whether a user will be able to create and change files in their home directory.

Printers Section

- The [printers] section is used in a manner similar to the [homes] section.

- If the [printers] section is present, a user can connect to any printer defined in the UNIX host's ***/etc/printcap*** file, even if the printer does not have a service entry in the ***smb.conf*** file. A sample configuration file:

```
[printers]
  comment = All Printers
  path = /var/spool/samba
  browseable = no
# Set public = yes to allow user 'guest account' to print
  guest ok = no
  writable = no
  printable = yes
```

- The "comment", "browsable," and "create mode" parameters are the same as you've just seen.

- The "printable" parameter simply tells Samba that this is a printer resource that you can print to. Setting the "public" parameter to "no" disables guest access and prevents unauthorized users from printing to printers on your system.

- The "writable" parameter is set to "no" because this is a printer resource, not a file system resource.

Sharing File Systems

- Once you've set up your system defaults for general operation, home directories, and printing, you can move on to setting up shared directories.

- Configuring a share with Samba is very easy: simply add a new section for the share you want to create, and fill in the parameters.

- The following example shares the files in the **/home/nfstest** directory with other Windows users.

```
[NFSHARE]
        comment = Windows Share to the NFS challenged
        path = /home/nfstest
        public = yes
        writable = yes
        guest ok = yes
        printable = no
```

- You must restart Samba for the changes to take effect. First enable the service as follows:

```
systemctl enable smb.service
```

- Then start it:

```
systemctl start smb
```

- Check the mount:

```
smbclient -L localhost
```

- When prompted for a password, simply hit enter. You should see everything your machine is offering as a SMB share.

User Authentication with SAMBA

- In order to access resources that have been shared with Samba, users need to be authenticated.
- User authentication allows Samba to restrict access to shares and to control read and write permissions for files and directories.
- Currently, Samba supports three different mechanisms for authenticating user access to shares, all of which are controlled by the "**security**" keyword in the *smb.conf file.*

Share Security

- The ***share*** security option tells Samba to restrict access based on **share permissions**. This is the oldest security model available in Samba and provides the least security. This option is now deprecated.

User Security

- The ***user*** security option tells Samba to use **username** validation for access to network shares.
- When a user attempts to connect to a share on a Samba server, Samba attempts to validate the user's username and password in the local password file.
- If the user is validated, the user is granted or denied access to the share based on its share permissions. This option is very useful if your PC-based accounts have the same usernames as your UNIX-based accounts.

- In order to use the user security option, you must create accounts on your UNIX system for all of your PC users that will be connecting to Samba shares.

Server Security

- Samba now allows you to redirect all user authentication requests to a separate SMB server. This SMB server does not have to be another UNIX system running Samba; it can be any SMB server capable of authenticating users, such as a Windows computer.

- So, by electing to use the server security model, you can centralize all your PC usernames and passwords on a Windows system and use that system for Samba authentication.

- In order to set up server-based security, add two lines such as the following to the [global] section of the *smb.conf file.*

```
security=server
password server = NTSRV1
```

- The first line tells Samba to use the server security model.

- The second line tells Samba which computer will act as the authentication server. You must specify the NetBIOS name for the authentication server, not its DNS name.

Restricting Access

- In addition to restricting access based on passwords (as explained above) we can also use the **xinetd** super daemon to run SAMBA and control access to shares.

- The following entries in the **/etc/xinetd.conf** file illustrate how to accomplish this:

```
service netbios-ns
{
    socket_type   = dgram
    protocol      = udp
    wait          = yes
    user          = root
    only_from     = 192.168.0.1 192.168.0.2
    group         = root
    server        = /usr/local/samba/bin/nmbd
}

service netbios-ssn
{
    socket_type   = stream
    protocol      = tcp
    wait          = no
    user          = root
    only_from     = 192.168.0.1 192.168.0.2
    group         = root
    server        = /usr/local/samba/bin/smbd
```