Samba

Aidan Fetter
10/16/2023

```
This document covers Samba and how to connect it to local
operating systems. You will need either two VMs for this or one
VM and one local machine (assuming they are connected to the same
NAT network).
```

1. What is the SMB protocol?

The Server Message Block (SMB) protocol is a TCP protocol created by Microsoft used to share access to files and printers across different nodes on and across (a) network(s), provide network browsing, and provide an authentication inter-process communication (IPC) mechanism.

- SMB now works on top of the TCP stack using port 445 but was originally used on NetBIOS (Network Basic Input/Output System) using ports 137-139
- Major vulnerability from SMBv1 – WannaCry ransomware (EternalBlue vulnerability)
- In a Microsoft Windows OS, SMB is implanted using the "LanmanServer" service, which is in charge of sharing the resources across the network, and the "LamanWorkstation" service, which is in charge of maintaining the host machine's name on the network
- Uses the Kerberos protocol (tickets) for larger networks and NTLM (New Technology LAN Manager) for smaller networks
  - Kerberos is protected by eavesdropping and replay attacks and uses symmetric-key cryptography (but may choose to use public-key cryptography)
- Newer versions of Samba use digitally signed messages to prevent against MITM attacks (default = do not allow unsigned connections)
- Up-to-date version: Samba v4.19

2. What is Samba?

- Samba is a software based off of the SMB protocol that allows Unix-like operating systems to communicate with Windows operating systems
- It also provides file and print services for Microsoft Windows clients using a Unix-based operating system
  - Netsmb is the SMB software uqsed for BSD systems (including macOS)

3. Samba Installation and Configuration

3.1. Install Samba and check for successful installation

Commands:

1. sudo apt install samba
2. Check if the installation was successful: whereis samba
3. Check to see if nmbd (the service that runs Samba) is configured: sudo systemctl status nmbd

3.2. Create a folder (or multiple folders) containing one or more files to share

Commands:

1. mkdir nameOfDir
2. cd nameOfDir && touch {a..z}.txt (or whatever command you want to use to add files to the directory)

3.3. Edit Samba configuration files

The configuration file is located in "/etc/samba/". Using whatever text editor you choose (vim, vi, nano, etc.) with root permissions, do the following:

1. Go to the bottom of the file and add the following lines:

```
[nameOfDir]
   comment = Samba in Linux #A description of the directory
   path = /home/user/nameOfDir #Path to your share directory
   read only = yes #The contents of the folder cannot be modified
   browsable = yes #Allows file managers to list the shared
directory under "Network"
```

3.4. Update your firewall to allow Samba

- Using ufw: sudo ufw allow samba
- Using firewall-cmd: sudo firewall-cmd --permanent --add-service=samba
- sudo firewall-cmd –reload

3.5. Add a new user to the Samba server

- sudo smbpasswd -a <name>
  - "<name>" should be the name of your user in your VM. For example, if your home directory contains the directory "user1", the command would be "sudo smbpasswd -a user1
  - You will then be prompted with a password. This can be any password you want and does not have to be the one you use for your VM. However, be sure to remember it as you will need it later when trying to access the shared files.

3.6. Start Samba and ensure connectivity

1. Start smb: sudo systemctl start smdb
   OR
   sudo systemctl start smbd
   - The two commands above start it on your current session
   - To enable it upon system startup: sudo systemctl enable smdb
     - Side note: Using "smb" instead of "smbd" also works. However, this document will use the latter for convenience
2. Start nmbd: sudo systemctl start nmbd

OR
sudo service nmbd start

3.7. Check if the two services are running properly:

- For smbd: sudo systemctl status smbd
  OR
  sudo service smbd status
- For nmbd: sudo systemctl status nmbd
  OR
  sudo service nmbd status

4. View shared files in Windows VM

4.1. Setting up network configurations

1. Locate your Windows VM and edit the "Network" settings to ensure that each OS resides on the same network. I did this by creating a new NAT network (essentially a LAN for VMs to interact with each other on their own network) and attaching it to each of the VMs). You can do this by clicking "Files" → "Tools" → "Network Manager" → "NAT Networks" → "Add".
   - If you're using VirtualBox or VMWare:
     When attaching the NAT network to each VM (or one if you chose to use your local host OS), ensure that "Promiscuous Mode" is set to either "Allow VMs" or "Allow All". This allows the VMs to "speak" to each other while they're on. You may need to save the state of the VM running the Samba server if needed; however, you should be able to change it while it's running.
2. Ensure that both smbd and nmbd are active and running on the Samba-hosting system. If you previously closed the VM, you might need to restart both services by typing "sudo systemctl start smbd" and sudo systemctl start nmbd".
3. Open the Windows VM and open the command line on both VMs. Run the command "ifconfig" on the Linux VM and "ipconfig" on the Windows VM to get their corresponding IP addresses. After obtaining this, ping the Linux machine from the Windows machine to ensure they are connected successfully (and vice versa). If not, go back to step 1 and review.
4. If both machines are connected to each other, press the Windows Key + R in the Windows VM to open up the Run application, and type "\\IP\Share", where "IP" is the IP address of the Linux machine and "Share" is the shared folder you created earlier. After pressing enter, you should now have access to the shared files after entering your credentials!

   - Side note: You can also run "\\IP" and the share will show up as the directory rather than the contents of the directory.

4. **View shared** files in Windows VM

4.2. If the IP address cannot be found and both machines are connected

1. Press the Windows Key + R in the Windows VM and type “regedit” to open up the Registry Editor. This will allow you to manage user and group access to specific system registries.

4. Follow the file path “HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\LanmanWorkstation\Parameters”, double click on “Parameters”, and change the value from 0 to 1

5. Open up the Run application once more and type “\\IP\share”, where “IP” is the IP address of the Linux machine and “Share” is the shared folder you created earlier.

6. You should now see the shared files on your Windows machine! Enter the credentials you made earlier to gain access to the files you created.

Example output: