

Introduction

Samba is a suite of tools for Linux (and other *nix systems) that speaks SMB the (Server Message Block), which is widely used as part of Windows network file sharing.

Our goal for this exercise is to set up a Windows file server on a virtual machine on your laptop.

Step 0 — Preparing your laptop's virtual machine

Use your existing virtual machine, or create a new one using the vlaunch script that we used at the last training.

If you're installing fresh, be sure to install the "openssh-server" package so that you can SSH into the guest. SSH to the Virtual machine. For these examples my virtual machine (aka guest) is called "oldmajor."

Step 1 — Installing samba.

In a terminal become root, either by logging in as root, or running the command

```
$ sudo -i
```

Then install the samba packages that you will need with:

```
0 root@oldmajor:~# aptitude install samba smbclient
```

This will get all of the packages you need to set up a windows file server. Once they have all downloaded, the installer will present you with a dialog that starts with

```
"Please specify the workgroup for this system.... Workgroup/Domain Name:"
```

This is the standard Windows "Workgroup" which may already be defined in your office. For this exercise, I went with "PIGPEN," but it you can choose any WORKGROUP name you like.

Step 2 — Adding directories and users

First create a directory (folder) that you want to share with Windows and make it group writable:

```
0 root@oldmajor:~# mkdir /srv/windowsshare
```

Then change the directory's group to Debian's default samba group, "sambashare," and make the directory group writable.

```
0 root@oldmajor:~# chgrp sambashare /srv/windowsshare/  
0 root@oldmajor:~# chmod g+w /srv/windowsshare
```

Now you want to create a user that can use samba

```
0 root@oldmajor:~# adduser smbtester  
Adding user `smbtester' ...  
Adding new group `smbtester' (1000) ...  
Adding new user `smbtester' (1000) with group `smbtester' ...  
Creating home directory `/home/smbtester' ...  
Copying files from `/etc/skel' ...  
Enter new UNIX password:  
Retype new UNIX password:
```

```
passwd: password updated successfully
Changing the user information for smbtester
Enter the new value, or press ENTER for the default
  Full Name []:
  Room Number []:
  Work Phone []:
  Home Phone []:
  Other []:
Is the information correct? [Y/n]
0 root@oldmajor:~#
```

The password you created in the previous step is for the user on the guest server; you will need to create a password that they can use with samba instead.

```
0 root@oldmajor:~# smbpasswd -a smbtester
New SMB password:
Retype new SMB password:
Added user smbtester.
0 root@oldmajor:~#
```

And lastly you will need to add this user to samba's group. Again, the default group in Debian is "sambashare."

```
0 root@oldmajor:~# adduser smbtester sambashare
Adding user `smbtester' to group `sambashare' ...
Adding user smbtester to group sambashare
Done.
```

Step 3 — Configuring Samba

Samba is a very flexible tool, so the configuration files can seem quite daunting at first. Don't worry, for most setups you can ignore most of the options, and just go with the defaults.

In Debian, the samba configurations are all stored in `/etc/samba/smb.conf`. This file is heavily commented, with lots of explanations about different options, so I'll only cover the changes you need to make for this example. Scroll to the bottom and add:

```
[SAMBA]
path=/srv/windowsshare
browseable=yes
writeable=yes
valid users = smbtester
admin users = root
```

Be sure that the path matches the directory you created, and the valid user is the username you just created.

Next, you will want to make sure that your Samba settings have the right permissions. That looks like:

```
0 root@oldmajor:~# testparm
Load smb config files from /etc/samba/smb.conf
rlimit_max: rlimit_max (1024) below minimum Windows limit (16384)
Processing section "[homes]"
Processing section "[printers]"
Processing section "[print$]"
Processing section "[SAMBA]"
```

```
Loaded services file OK.
Server role: ROLE_STANDALONE
Press enter to see a dump of your service definitions
```

...

```
[SAMBA]
path = /srv/windowsshare
valid users = smbtester
admin users = root
read only = No
```

Since you changed the configuration file, you will want to re-start the Samba service:

```
0 root@oldmajor:~# service samba restart
Stopping Samba daemons: nmbd smbd.
Starting Samba daemons: nmbd smbd.
```

Now you want to check and make sure that the configuration for your test user is actually right.

```
0 root@oldmajor:~# smbclient -L //localhost -U smbtester
Enter smbtester's password:
Domain=[PIGPEN] OS=[Unix] Server=[Samba 3.5.6]

  Sharename      Type      Comment
  -----      -
  print$         Disk     Printer Drivers
  SAMBA          Disk
  IPC$           IPC      IPC Service (oldmajor server)
  smbtester      Disk     Home Directories
Domain=[PIGPEN] OS=[Unix] Server=[Samba 3.5.6]

  Server          Comment
  -----
  OLDMAJOR        oldmajor server

  Workgroup       Master
  -----
  PIGPEN          OLDMAJOR
0 root@oldmajor:~#
```

Step 4 — Testing it all out.

Back on your host machine (not your virtual guest machine) make sure that you have the "smbclient" package installed on your laptop (use synaptic, or just "aptitude install smbclient"). There's a good chance that it may already be there.

Next open up a file browser window and select the Network location, and drill down to the test setup that you just created. Here you will be prompted for your user name and password; this is the samba password that you created earlier. You will then be presented with a blank folder, into which you can drag and drop a file. If that works, you're done!

Alternately, you can also test this on the command line from your host with this command (change your IP address and service name as needed), the service name I'm using is SAMBA (the same as from the

configuration above), and of course the user is "smbtester" for this example.

```
0 nat@pigtown:~$ smbclient //10.11.13.40/SAMBA -U smbtester
```

You will be asked to enter your samba password. If you are successful this will drop you into a funny looking shell:

```
0 nat@pigtown:~$ smbclient //10.11.13.40/SAMBA -U smbtester
Enter smbtester's password:
Domain=[PIGPEN] OS=[Unix] Server=[Samba 3.5.6]
smb: \>
```

I will spare you the details of using command line FTP style shells, for testing purposes, all you need to do is create a directory, and see if it everything works.

```
smb: \> mkdir test-directory
smb: \> ls
.                D            0   Thu Jan 10 00:27:32 2013
..               D            0   Wed Jan  9 23:28:03 2013
foo              A            0   Thu Jan 10 00:12:35 2013
test-directory  D            0   Thu Jan 10 00:27:32 2013

    34386 blocks of size 131072. 26179 blocks available
smb: \> exit
```

If you got no errors, that's it! You have a working windows file server running entirely on your Linux virtual machine!