**Lesson 5**

**Samba**

**Contents**

[1. Creating a Samba Share 2](#_Toc91959030)

[2. Defining Samba users for standalone Samba server 3](#_Toc91959031)

[3. Browsing Samba Shares from the Client 7](#_Toc91959032)

[4. Accessing anonymous Samba Shares from the Client 8](#_Toc91959033)

[5. Only allow authenticating users to access Samba Shares 10](#_Toc91959034)

[6. Uploading files to Samba Share 12](#_Toc91959035)

[7. Mounting Samba Share automatically upon bootup 14](#_Toc91959036)

[8. Accessing Home Directories through Samba 17](#_Toc91959037)

[9. Encrypting the smb traffic 18](#_Toc91959038)

[10. Accessing Samba share from a Windows System (Optional) 20](#_Toc91959039)

1. Creating a Samba Share

"Samba is a free software re-implementation of the SMB networking protocol. Samba runs on most Unix, OpenVMS and Unix-like systems, such as Linux ...."

SMB networking protocol provides network file and print services.

~Ref: https://en.m.wikipedia.org/wiki/Samba\_(software)

On server:

1. Install the Samba server and the Samba client utilities (eg: smbpasswd):

dnf install samba samba-client

If you only need to install the client utilities you may use the following:

dnf install samba-client

1. Start the Samba service and make it start automatically on bootup.

systemctl start smb

systemctl enable smb

1. Create a directory for sharing. Change its group owner to the group ppm [your should have created this ppm (peter, paul and mary) group in the previous practical].

mkdir /samba\_share

chgrp ppm /samba\_share

1. Give full access to the group owner.

chmod g=rwx /samba\_share

or

chmod 0775 /samba\_share

Text

Description automatically generated with medium confidence

1. Create a file called sambafile1.txt with the following content in the /samba\_share directory.

An SMB exploit is a fairly common cyberattack.

In Windows systems before Windows 10, the most famous SMB attack is WannaCry.

This attack exploited the overflow vulnerability with the EternalBlue

exploit and persisted worldwide in systems for a year and a half.

Another example of the EternalBlue exploit is Emotet, which targets banks.

Other SMB exploits include EternalRomance, used for NotPetya and Bad Rabbit,

and EternalEnergy. There is the possibility of a fourth exploit called

EternalSynergy,

In Windows systems before Windows 10,one particular vulnerability stands out

from the crowd: CVE-2020-0796. This is a critical vulnerability in the Server

Message Block (SMB) protocol in new versions of Windows operating systems.

This SMB vulnerability could cause a wide range of wormable attacks and

potentially a new Eternal Blue. Without going into the gory details,

a flaw in the new SMBv3 compression mechanism potentially allows an attacker

to take down or take over a Windows system.

reference:

1. https://www.paladion.net/blogs/protection-againsttheserver-message-block-smb-vulnerability-exploit-paladion.

2. https://www.guardicore.com/2020/03/how-to-protect-your-systems-against-critical-smb-vulnerabilities-cve-2020-0796/

A screenshot of a computer

Description automatically generated with medium confidence

1. Defining Samba users for standalone Samba server

In the windows platform, SMB is implemented via shared folders. The access control of the shared folders (or shares) is based on the windows authentications (ie. NTLMv2 or Kerbebors).

The Samba implementation in Linux, there are two different access control schemes.

The simple one, standalone server scheme, requires to setup an additonal samba user database. To enable a user to access to the samba, there are two prerequisites.

The user must be created in the Linux system, and this user must be added to the samba user database.

The other one, domain control scheme, requires some additonal system configuration to let the Linux server to join to an Active Directory Domain and the user database will be integrated with the Active Directory User database. We will not cover this scheme in ST2412.

1. Edit the file /etc/samba/smb.conf. Append the following to the end of the file.

[myshare]

comment = My Samba Share for peter , paul and mary

path = /samba\_share

guest ok = yes

Text

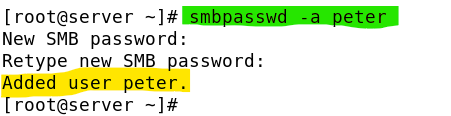
Description automatically generated

The [myshare] block defines the access control configuration for the /samba\_share. It is set to allow anonymous login (No user id nor password required).

By default, valid samba users can also access to this share. At this point, there is no samba user configured yet.

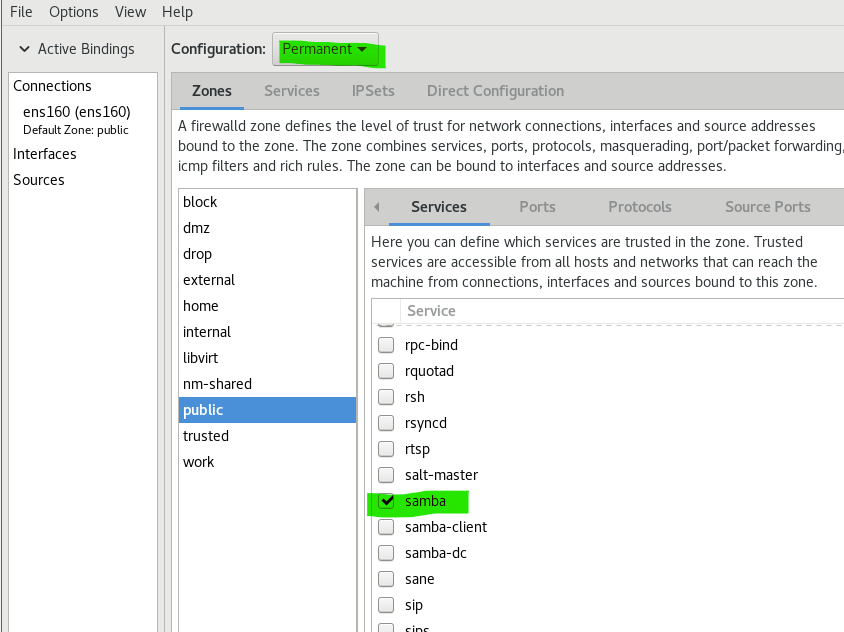
1. Add a samba password for user “peter”.

smbpasswd -a peter

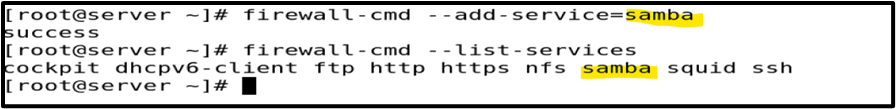


Take note 'peter' must be an existing user in the system. that the smb password does not need to be the same as peter's Linux password. Suggest to set the SMB password of peter as 'peter' for this simple exercise.

1. Adjust the firewall to allow connections to your Samba server. [using runtime or permanent].



You may use firewall-cmd to allow samba servcie too.



1. Defining host name for name to IP resolution

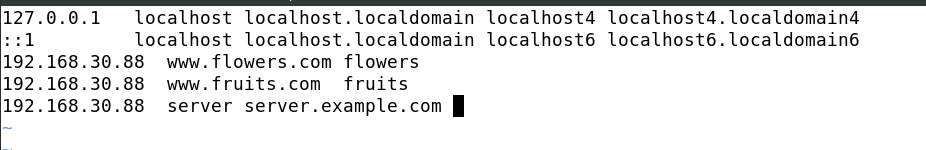
The Samba server will normally try to resolve its hostname, and as we have not setup any DNS service yet, you may update the /etc/hosts file which records some static host reference to provide the such name resolution for the time being.

Add an entry of your server to the /etc/hosts

e.g.

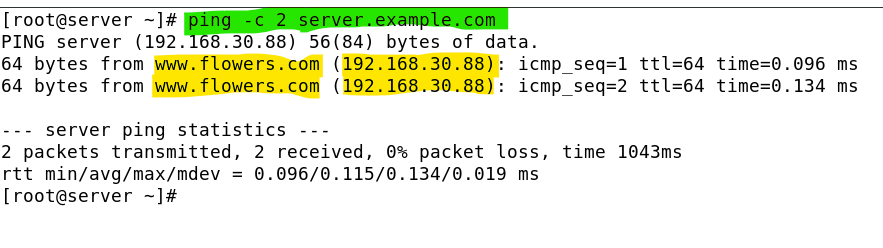
192.168.30.88 server server.example.com

\*Take note that you need to replace 192.168.30.88 with your actual server IP.



The new entry

You may test your host file settings by pinging your own server with 'server.example.com':



Take note that, in the above sample screen shot, it reports the targeted host is [www.flowers.com](http://www.flowers.com). It is because the host (192.168.30.88) has associated to multiple host names, and the [www.flowers.com](http://www.flowers.com) is the first one in the /etc/hosts file.

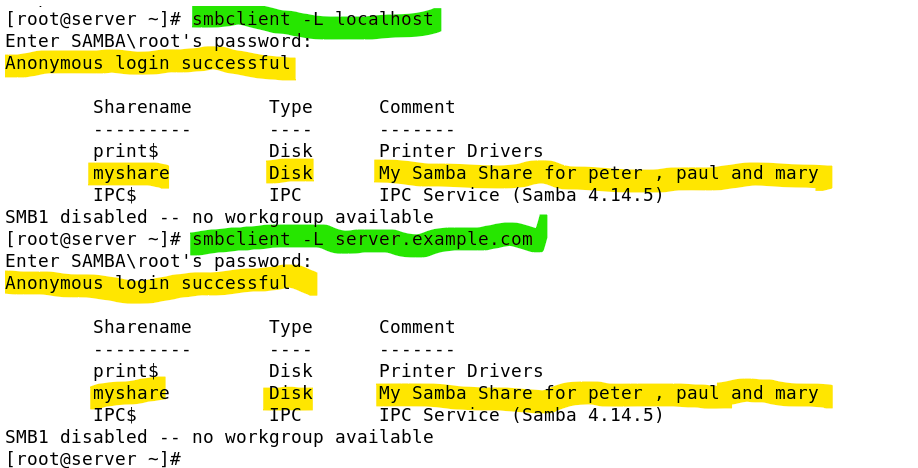
1. Browsing Samba Shares from the server

Remain login as root, run the following command to browse the available Samba Shares offered by the server:

smbclient -L localhost (or smbclient -L server.example.com )

1. Press enter when asked for root's password.

'In this case (you did not provide any password), you are trying to access to the server as an anonymous user.'



Note: Have you adjusted the server firewall to allow smb service? or else you may encounter the connection failed issue.

1. Browsing Samba Shares from the Client

On client (login as student) :

1. Install the Samba client.

sudo dnf install samba-client

Note: Have you updated the /etc/sudoers file to allow student to run commands with sudo ?

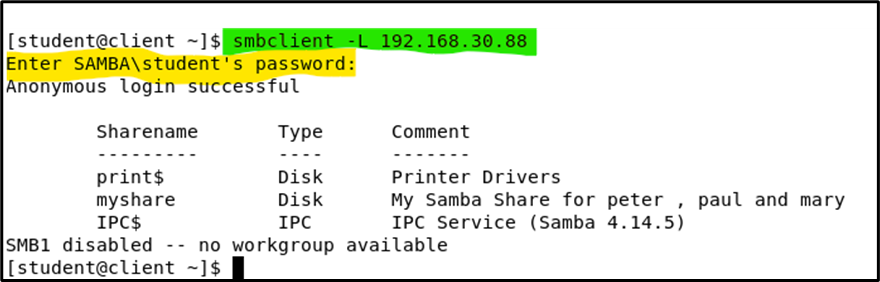
Have you disable the local repo (based on the server ftp) and resume the online repo ? For the rest of this semester, it is recommended to use the Oracle online repo rather than your own local server repo.

1. View the Samba shares on the server.

smbclient -L <*serverIP>*

**Press enter** when asked for student's password.

* 1. 'In this case (you did not provide any password), you are trying to access to the server as an anonymous user.'



Note that myshare is listed among the Sharename section.

On server:

1. Edit the file /etc/samba/smb.conf. Add the line (highlighted in bold) to turn off browsing.

[myshare]

comment = My Samba Share for peter , paul and mary

path = /samba\_share

guest ok = yes

**browsable = no**

On client:

1. View the Samba shares on the server again. This time, **myshare is not listed**.

smbclient –L *serverIP*

Table

Description automatically generated with medium confidence

1. Accessing anonymous Samba Shares from the Client

On client (login as root or student):

1. Access the samba share as an anonymous user.

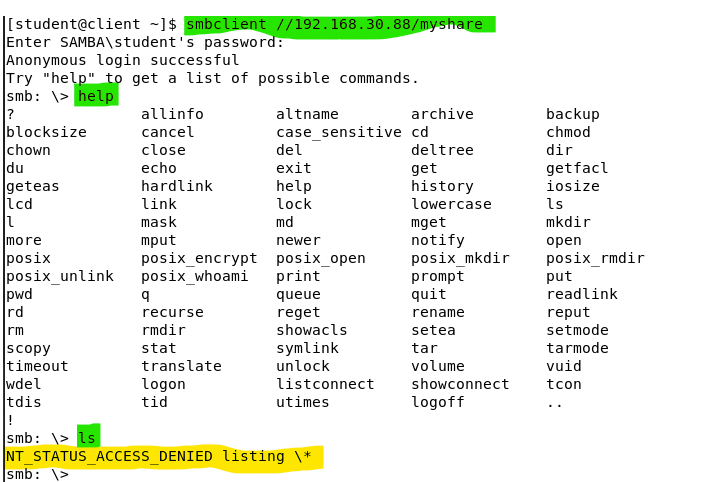
smbclient //*serverIP*/myshare

or

smbclient //serverIP/myshare

1. **Press enter** when asked for the root / student password.
2. At the Samba prompt, type “help” to view the available commands.
3. Type “ls” to list the contents of the shared directory.

If you get an error, it could be due to the SELinux setting on the Samba Server.



On server:

1. Turn off SELinux

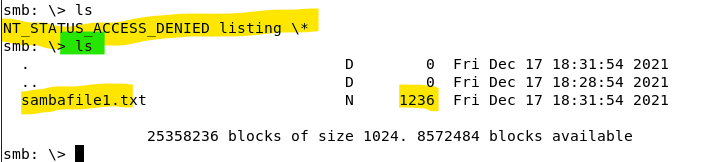
setenforce 0

On client:

1. Repeat the command that had caused the error previously.

If it works now, you know the previous error is due to SELinux.

You may enter the 'exit' command to quit the smbclient shell.



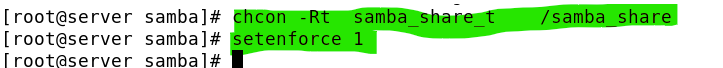
On server:

1. The error could be due to the /samba\_share directory having the wrong context. Set the context to samba\_share\_t.

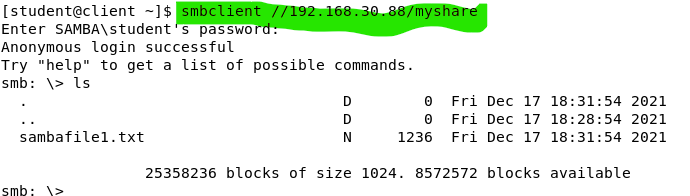
chcon -Rt samba\_share\_t /samba\_share

1. Turn on SELinux again using :

setenforce 1



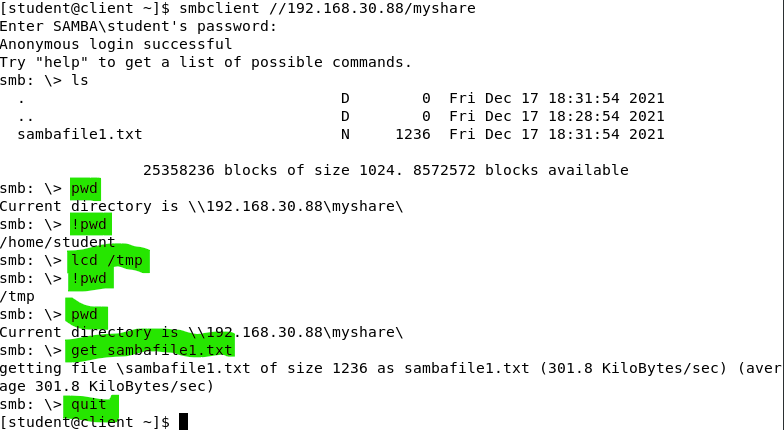
Now at the client you should be able to achieve the following:



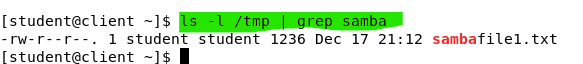
On client:

1. Remain in the Samba Client login session:

* Type 'pwd' to check your current folder at the server.
* Type '!pwd' to check your current folder at the client. (local pwd)
* Type 'lcd /tmp' to change your current folder to /tmp at the client. (local cd)
* Type '!pwd' to check your current folder at the client.
* Type 'get sambafile1.txt' to download the shared file.
* Type 'quit' or 'exit' to exit from the Samba client.



1. Verify that you have downloaded sambafile1.txt into your /tmp folder.



1. Only allow authenticating users to access Samba Shares

Before you proceed into this section, ensure you have completed the Section 1 – Step 2 instruction : " smbpasswd -a peter"

On server:

1. Edit the file /etc/samba/smb.conf. Remove or comment out the line “guest ok = yes” from the [myshare] section.

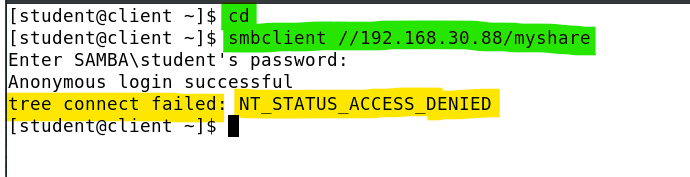
Text

Description automatically generated

On client:

1. Try to access the samba share as an anonymous user. You should not be successful.

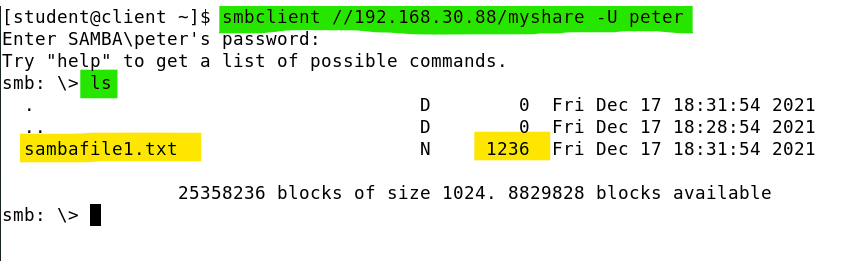
smbclient //*serverIP*/myshare



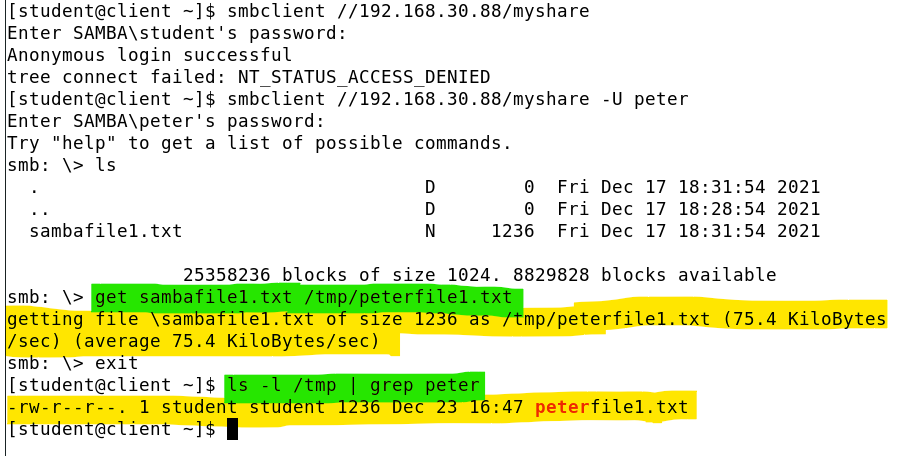
1. Access the samba share as user peter.

smbclient //*serverIP* /myshare –U peter

Enter the Samba password for peter.



1. Verify that peter can download the sambafile1.txt. See how it can be done in the following screenshot:



1. Verify that 'paul' is not able to use smbclient to access to the myshare from the client machine as you did not prepare the smbpasswd for paul. [Optional]
2. Uploading files to Samba Share

On client: (login as student)

1. Create a file for uploading:
   * + - 1. echo "Hello world" > hithere.txt
2. Access the samba share as user peter.

smbclient //*serverIP*/myshare –U peter

1. Enter the Samba password for peter.
2. Type "put *local\_filename target\_filename*" (replace *local\_filename* with the name of your file, e.g. 'hithere.txt') to upload the file to the Samba share. You **should not be** successful as the Samba share is not configured to be writeable by default.

Text

Description automatically generated with medium confidence

1. Type “exit” to quit from the smb shell.

On server:

1. Edit the file /etc/samba/smb.conf. Add the following line in bold to allow the user group ppm to modify files. [ or you can just allow the user 'peter']

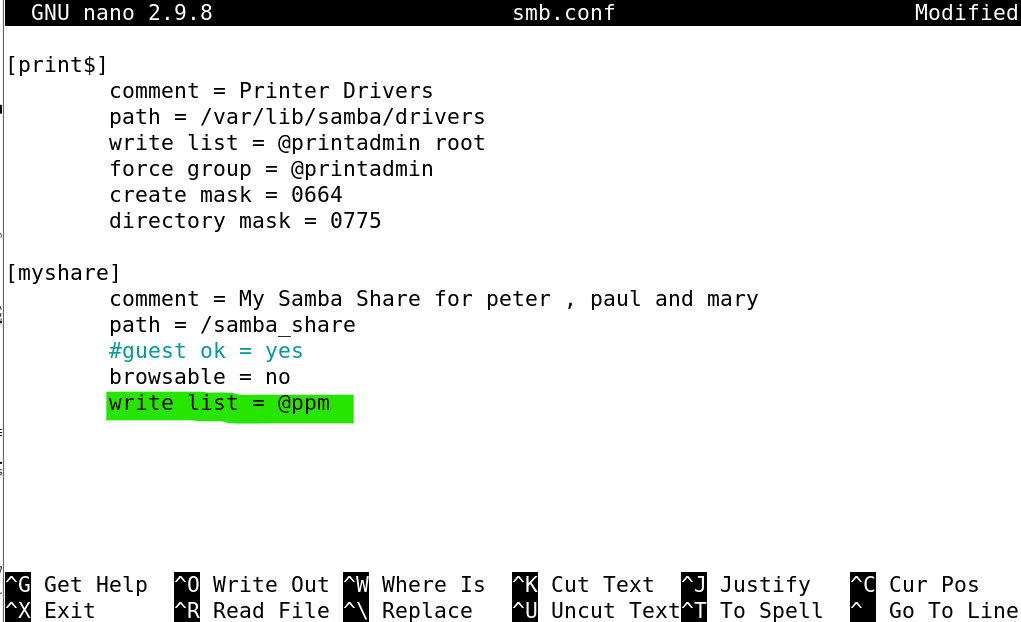
[myshare]

path = /samba\_share

#guest ok = yes

browsable = no

**write list = @ppm**

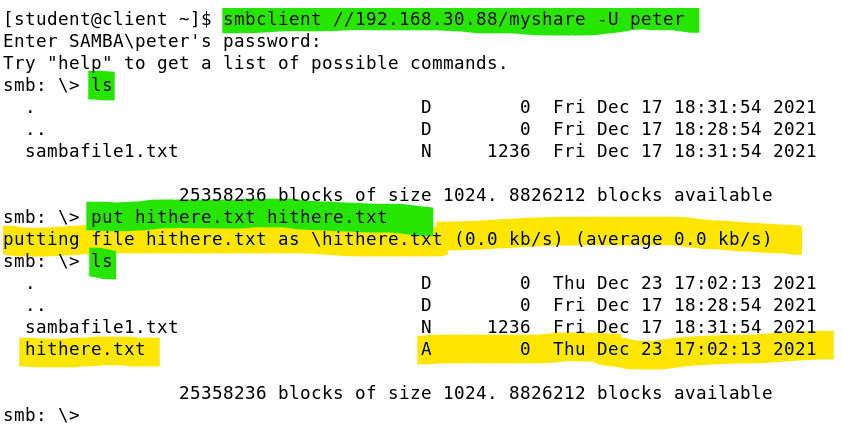


On client:

1. Access the samba share as user peter.

smbclient //*serverIP*/myshare –U peter

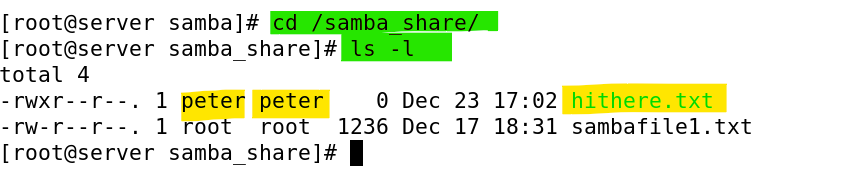
1. Enter the Samba password for peter.
2. Type "put *local\_filename target\_filename*" to upload the file to the Samba share. You should be successful this time.



1. Type “exit”.

On Server:

1. Verify the file has been uploaded successful and check the ownership of the file.



1. Mounting Samba Share automatically upon bootup

If we want to open the samba shares to al the users of a client system. We can mount the share to the remote client system using one of the valid user credentials. However, all the file operations will be committed by the same user account.

On client:

1. As user root (or using sudo) , install the cifs-utils so that your client can mount the Samba share using the Common Internet File System (CIFS)\*.

dnf install cifs-utils -y

\* "Server Message Block (SMB) v.1.0 protocol was rebranded as Common Internet File System (CIFS)since 1990's. " - ~ https://cifs.com/

1. Create a mount point /sambadata.

mkdir /sambadata

1. Verify if you can mount the Samba share (myshare) to your sambadata:

Use the mount command with the -t and -o options to try the mounting is possible:

mount -t cifs -o username=peter //<server IP>/myshare /sambadata/

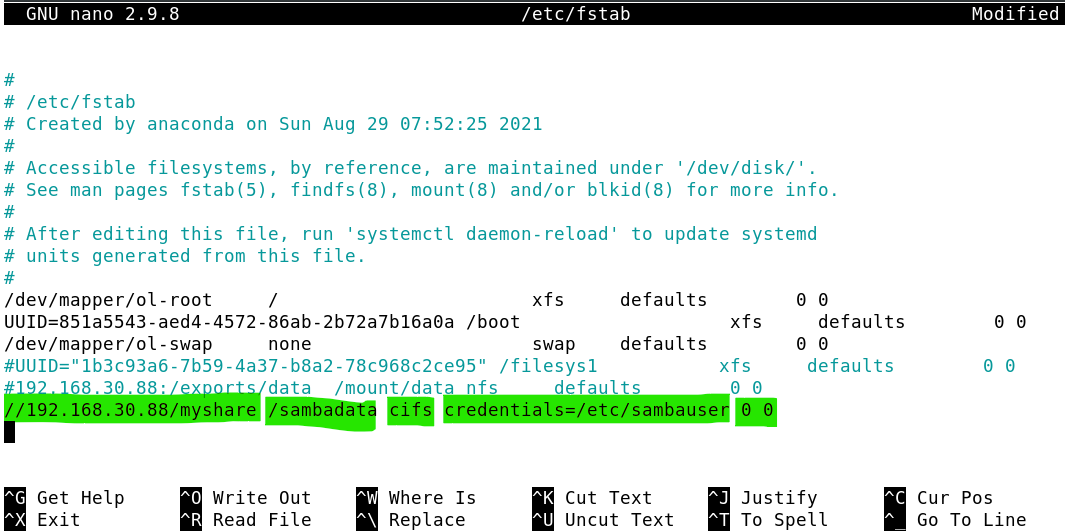
Text

Description automatically generated

Remember to use the 'umount' command to unmount the share.

1. Edit the file /etc/fstab and add the following line.

**//*serverIP*/myshare /sambadata cifs credentials=/etc/sambauser 0 0**



1. Create a new file /etc/sambauser and enter the login credentials.

user=peter

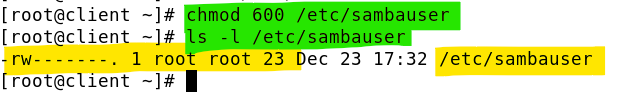
Change to the value of the Samba password you set for samba user peter

pass=<peterpassword>



1. Secure the file containing the login credentials. (disable others to read)

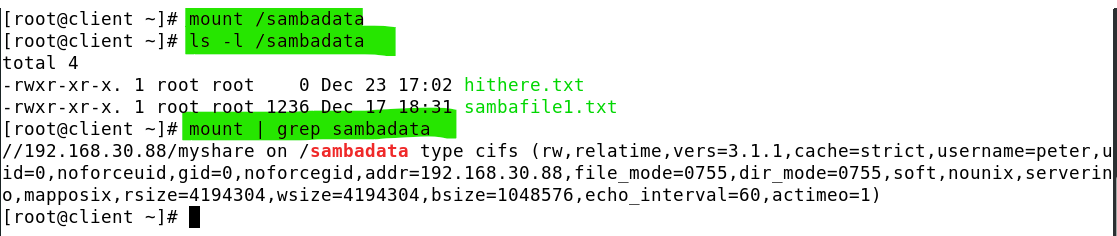
chmod 600 /etc/sambauser ( or chmod u=rw, g=, o= /etc/sambauser)



1. Mount the exported directory as specified in /etc/fstab.

mount /sambadata

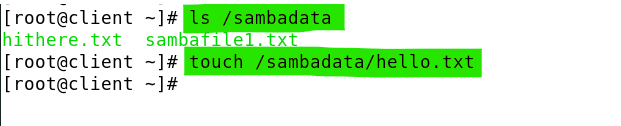
and view if there is any files listed in the /sambadata after the mount.



On client (Login as root)

1. Create a new file, hello.txt in /sambadata .

touch /sambadata/hello.txt



List out the file details in /sambadata :

ls -l /sambadata

Who are the owners of the files in this folder?

On Server:

1. View the contents of /samba\_share.

ls -l /samba\_share

What do you observe regarding the content listed in the /samba\_share and the /sambadata at the client side ?

On client:

1. When you have completed this test, you can comment out the line you added to /etc/fstab so that the samba share won’t be automatically mounted the next time you restart the client.
2. You may unmount the samba file system now.

umount /sambadata

1. Accessing Home Directories through Samba

So far, we only use samba to share out a specific local folder to the remote client. With the authentication enabled, we can let remote client to access to their corresponding user home folders. I.e. When peter access to the server via samba, he can access to his own home folder /home/peter.

On server:

1. Check that /etc/samba/smb.conf contains the following lines to share home directories.

[homes]

comment = Home Directories

valid users = %S, %D%w%S

browseable = No

read only = No

inherit acls = Yes

[Note: For the valid users entry:

%S maps to the service name (in this case is the user id).

%D maps to the Domain of the system (in our case, we do not have Domain to be involved)

%w maps to the winbin seperator e.g. \

In the above setup, each Home Directory allows two possible users:

Let's use peter as an example:

peter , domain\peter ]

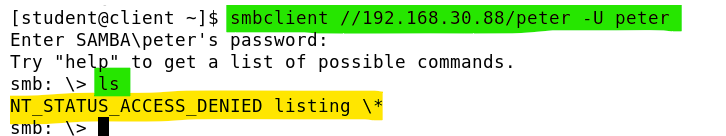
On client:

1. Access the home directory through samba share as user peter. Type:

smbclient //*serverIP*/peter –U peter

1. Enter the Samba password for peter.
2. Type “ls” to view the contents of peter’s home directory on the server.

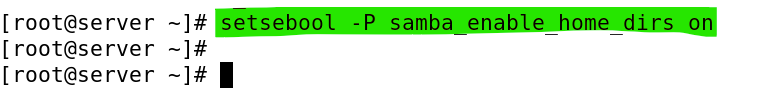
You should be able to login but you cannot really access to the content of peter's home directory due to the SELINUX restrictions.



On Server:

1. Turn on the SELinux Boolean samba\_enable\_home\_dirs (make the setting persistent).

setsebool -P samba\_enable\_home\_dirs on

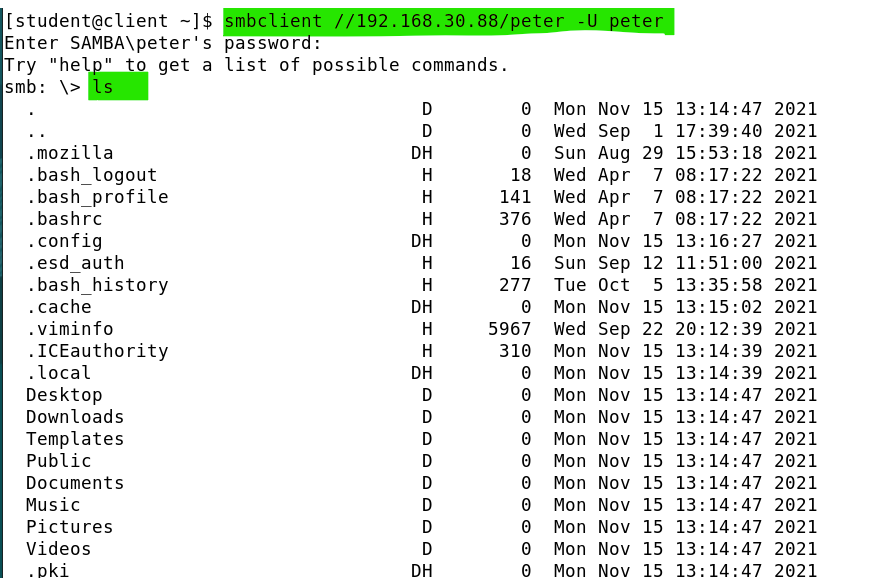


On Client:

1. retry step 2 - step 4. To login as peter to his home directory and list the content.

This time should work.

1. Try get and/or put files from and to the folder.



1. Type "exit" to disconnect.

Your task: Do the necessary to enable paul and mary to access their home directories and the /samba\_share from the client (login as student).

1. Encrypting the smb traffic

So far, we have not configured to require the client side / server side traffic to be encrypted. By default, the traffic is actually sent and received in plaintext. If you are using wireshark you can see all the content of the files that have been sent/retrieved via smb.

To secure the data transmission for integrity check and privacy, we need to add in a few parameters in the global section of the smb.conf file.

On server:

* 1. Edit the /etc/samba/smb.conf, add the line (highlighted in bold) to the global section.

[global]

workgroup = SAMBA

security = user

passdb backend = tdbsam

printing = cups

printcap name = cups

load printers = yes

cups options = raw

**min protocol = SMB3**

**client signing = mandatory**

**client smb encrypt = required**

**server signing = mandatory**

**server smb encrypt = required**

Text

Description automatically generated with medium confidence

* 1. Run the command , testparm, to check the current effective smb configuration settings.

testparm

Graphical user interface, text, application

Description automatically generated

Observed that the parameters listed by testparm is different from the smb.conf file.

* 1. If you do not see some of the added parameters are missing (not yet in effect), you need to restart the smb service, type:

systemctl restart smb

* 1. Rerun testparm to verify.
  2. Try to run smbclient and wireshark to verify the encryption is really working. [Optional]

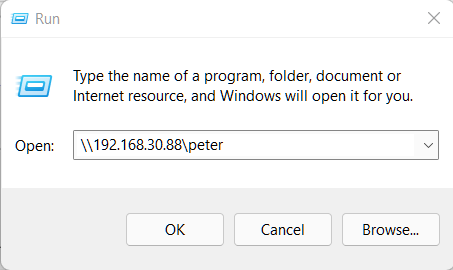
1. Accessing Samba share from a Windows System (Optional)

By default, your own notebook can connect to all your VMWare VMs\*. Thus, you should be able to access to the samba shares that hosted in the LAS server VM.

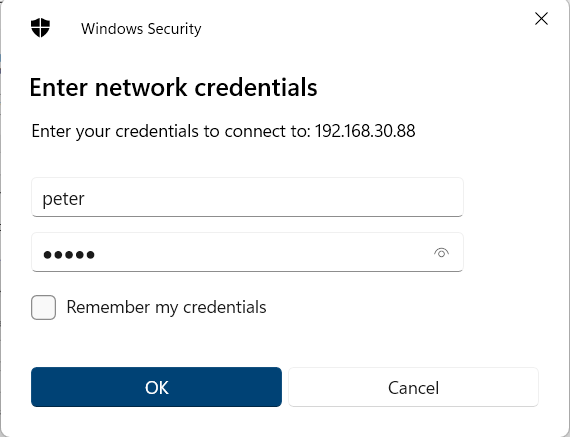
\*Your firewall/endpoint protection tools on your own notebook may block such connections.

On any Windows system (eg your Base PC):

1. Go to Start, Run.
2. Type \\*serverIP*\peter (or \\serverIP\myshare) and click OK.



1. Login as peter and enter peter’s Samba password.



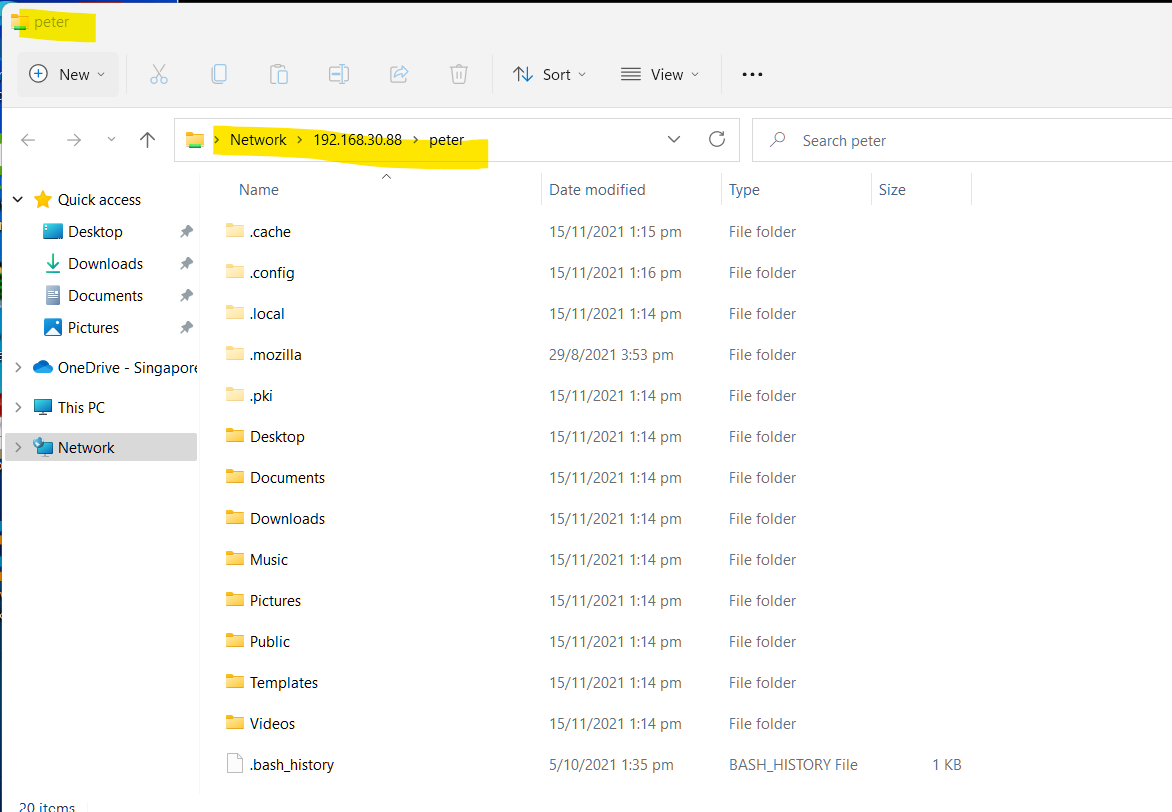
You may need to do an extra step if your PC is currently logged in to a Windows Domain, your PC will try to use the domain user account to login for you. In this case, you need to override this default behaviour. (See the sample below)

Graphical user interface, application

Description automatically generated

Click on More Choices to access to the Use a different account option to override the default user name and password.

1. You should be able to see the contents of the Linux share. You should be able to read and copy files to the peter's home directory from your Windows system.



*End of Practical*