**TP B**

**Sistemas Operativos**

**Conexión entre SO distintos**

Para esta actividad utilizaremos un sistema operativo Ubuntu 18.04 y un sistema Windows 10

**Instalar Samba**

Para compartir archivos instalaremos un servidor Samba en el SO Ubuntu, por medio del comando

*apt install samba*

*,* para instalar el servidor Samba.

**Crear o definir usuarios**

Podemos crear usuarios específicos para Samba, o podemos generar una contraseña para uno o más usuarios existentes.

Crear un usuario:

*useradd sambauser*

o usamos un usuario existente, esteban

luego generamos la contraseña para el usuario en samba

*smbpasswd -a* sambauser

ó *smbpasswd -a* esteban

le colocamos la contraseña, que debería ser distinta a la del mismo usuario, si se utiliza uno existente, para su acceso, verificamos los usuarios de Samba con el comando

*pdbedit -L*

Con el comando

*smbpasswd -x* USER podremos eliminar un usuario de Samba

**Usuarios creados**

Para pruebas creamos tres usuarios

suser : usuario de lectura en todos los directorios. Podra escribir y ejecutar en *public* y *lectura*

USERPASS: suse

SAMBAPASS: susamba

sreader: solo tendrá permisos de lectura en los directorios *lectura,*y no podrá acceder a ningún otro directorio

USERPASS sure

SAMBAPASS: resamba

swrite: podrá leer escribir en la carpeta *soloAutWrite* y *public,* y podrá acceder a el directorio *lectura*

USERPASS: swri,

SAMBAPASS: wrsamba

nobody: usuario anonimo, ya existente en el sistema UNIX, este usuario es el usuario anonimo, por lo que podrá ver que existen los directorios, pero no podrá ver ningún dato de ellas

**Creación de directorio locales**

Podemos utilizar directorios ya existentes o generar espacios específicos para compartir.

Crear las carpetas locales, soloLectura, publico, soloAutWrite

Mkdir /samba/FOLDER

**Creación de grupo de usuarios**

creamos el grupo admines, escribe y validos

groupadd admines

groupadd validos

groupadd escribe

agregamos a cada usuario al grupo correspondiente

usermod -G admines,validos swrite

usermod -G validos,escribe suser

usermod -G validos sreader

cambiamos el grupo de los directorios del grupo root, a el grupo que estará habilitado para modificar, con la opción -R, nos aseguramos que la misma configuración de grupo se aplique a los archivos y directorios existentes, como a los nuevos que se crearan

chgrp -R admines /samba/soloAutWrite

chgrp -R validos /samba/publico

chgrp -R escribe /samba/soloLectura

Cambiamos los permisos del directorio, diciendo que se aplicara a todos los nuevos directorios y archivo (-R), 0775, significa en octal, el 0 es indicador de la base, el primer 7, en octal, indica que el propietario, es decir el usuario ROOT, podrá leer, escribir y ejecutar cualquier archivo o directorio, el segundo 7 indica que cualquiera perteneciente al grupo correspondiente, podrá realizar lectura, escritura y ejecución, y el ultimo 5 indica que cualquier usuario no perteneciente a el grupo solo podrá leer y ejecutar. La opción 0774, en soloAutWrite, permite que el dueño y el grupo admines puedan realizar toda acción, pero cualquier otro usuario solo podrá leer.

chmod -R 0774 /samba/soloAutWrite

chmod -R 0775 /samba/soloLectura

chmod -R 0775 /samba/publico

**Configuración de samba**

Ahora debemos configurar el Samba, modificaremos el archivo smb.conf, ubicado en etc/samba/

Modificaremos la configuración global del Samba

Agregaremos el identificador del servidor, el nombre del workgroup, y el nombre que tendrá el servidor

[global]

## Browsing/Identification ###

netbios name= so-tpb

# Change this to the workgroup/NT-domain name your Samba server will part of

workgroup = WORKGROUP

# server string is the equivalent of the NT Description field

server string = %h - Sistema servidorS

debemos indicar en que red esta escuchando, como es un servidor local añadiremos la ip/mask privada

#### Networking ####

# The specific set of interfaces / networks to bind to

# This can be either the interface name or an IP address/netmask;

# interface names are normally preferred

; interfaces = 127.0.0.0/8 eth0

*interfaces = 192.168.1.0/24 eth0*

Podemos definir que rol tendra el servidor, standalone server, member serve, classic primary domain controller

####### Authentication #######

# Server role. Defines in which mode Samba will operate. Possible

# values are "standalone server", "member server", "classic primary

# domain controller", "classic backup domain controller", "active

# directory domain controller".

#

# Most people will want "standalone sever" or "member server".

# Running as "active directory domain controller" will require first

# running "samba-tool domain provision" to wipe databases and create a

# new domain.

server role = standalone server

lo dejaremos en *standalone server*

Pondremos que la seguridad, necesitara un usuario y contraseña para el acceso, esto es redundante, ya que el rol del servidor standalone, toma este parámetro, pero nos aseguraremos que se utilice esta opción

####### Authentication #######

.

.

Security = user

En la sección donde definiremos los directorios compartidos tenemos diferentes parámetros que podemos definir para el directorio compartido, lo que se debe tener en cuenta es que el control sobre escritura, lectura lo hace el mismo sistema operativo Linux, por lo que debemos generar los usuarios, grupos y permisos, sobre los archivos o directorios para el correcto funcionamiento

Estructura básica

[SHARENAME] nombre con el que se conocerá el directorio, publicamente

**Path**= PATH/TO/FOLDER, ubicación del directorio local

Comment= INFO, información sobre el archivo compartido

Read only= yes/no, será de solo lectura o permitirá escritura

Guest ok = yes/no, permitirá el acceso de usuarios anónimos

Browseable = yes/no, permitirá que al acceder a la dirección ip se vea el directorio compartido

Valid users= USER, @GROUP; indica que usuario o grupo de usuarios tiene permitido acceder al directorio

Invilid users= USER,@GROUP; indica que usuario o grupo de usuarios no podrán acceder al directorio

Write list= USER,@GROUP; indica que usuario o grupo de usuarios podrán escribir específicamente, esto es en el caso de read only = yes

Create mask = [0775], define el valor por defecto de los atributos que tendrán, los archivos generados en este directorio, es decir, que el usuario y el grupo al que pertenece podrán realizar toda acción, pero los demás usuarios solo podrán leer y ejecutar

Directory mask=[0700], define el valor por defecto de los directorios que se creen en este directorio, es decir, solo el creador podrá realizar acciones sobre los directorio generados

Ahora agregaremos los recursos a compartir en smb.conf, crearemos una carpeta llamada **public**, que apunta al directorio local **/samba/publico**

[**public**]

Path= **/samba/publico**

Comment= carpeta publica a todos

Read only =no

Guest ok = no

Browseable = yes

Invalid users = sreader

Write list = @validos

Esto define un directorio llamado public, que permitirá lecturas y escrituras, no permitirá el acceso anónimo, toda persona que ingrese a la dirección ip vera el directorio, y sreader no podrá acceder a los archivos de este directorio, pero si podrá ver que existe el mismo

Ahora configuraremos una carpeta de solo lectura, es decir solo los miembros del grupo **escribe** podrán escribir en ella, y los otros usuarios podrán solo ver, a este directorio le llamaremos **lectura**, apuntando a el directorio local **\samba\soloLectura**

[**lectura**]

path=/**samba/soloLectura**

comment= carpeta publica a todos

read only = yes

guest ok = no

browseable = no

valid users= @validos

write list = @escribe

A este directorio no se podrá acceder siendo un usuario anónimo, solo podrán acceder los usuarios definidos en Samba, todos pertenecen al grupo validos, y los miembros del grupo escribe serán los únicos que podrán escribir.

Ahora configuraremos la carpeta que solo el dueño y los miembros del grupo admines puedan modificar y ver, y el usuario suser podrá acceder pero solo podrá leer los datos

[**paraAdmin**]

path=/**samba/soloAutWrite**

comment=solo admin y root

read only = no

guest ok = no

browseable = no

valid users = @admines, suser

write list = @admines

Reiniciamos el servicio, para que los cambios tomen efecto

Systemctl restart smbd.services

ó

service smbd restart && service nmbd restart

Ahora iremos a Windows

Podemos ejecutar la llamada a alguna de las carpetas ejecutando una nueva tarea

Windows+R

Y escribiendo la dirección ip con la carpeta especifica, esto nos dará un acceso a la carpeta pero no agregara la ubicación al sistema de archivos

\\IP\SHAREFOLDER

O en el explorador de archivos ingresando la ip, esto nos mostrara todas las carpetas que tengan la bandera browseable = yes

\\IP

\\IP\SHAREFOLDER

Esto nos dará acceso al sistema de archivos compartido, pero no tendremos un enlace permanente al servido, es decir, siempre deberemos indicar el path con su dirección ip y/o carpeta específica para acceder.

Para agregar un enlace permanente, en Windows 10, debemos estar en el equipo agregar una unidad de red y especificar el ip o dominio y la carpeta especifica a la que queremos vincular

\\192.168.1.17\public

\\192.168.1.17\paraAdmin

\\192.168.1.17\lectura

Para acceder debemos ingresar las credenciales necesarias, y se obtendrá el acceso y permisos definidos.

En esta experiencia, todas las carpetas serán visibles en el explorador de archivos, pero exigirá credenciales para acceder a los diferentes directorios. Si se hubiera definido browseable = no, deberemos indicar el path correspondiente para acceder al recurso.

Las carpetas a las cuales se definió guest ok = no, exigirán que se utilice el usuario y contraseña, existente en el servidor Samba para su acceso, pero aquellos usuarios o grupos de usuarios sin los permisos necesarios no podrán leer, escribir ni ejecutar ningún archivo o directorio.

|  |  |
| --- | --- |
| User | Group |
| Suser | Escribe, validos |
| Sreader | Sreader, validos |
| Swrite | Admines, validos |
| nobody | Nogroup |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Directorio/Usuario | suser | sreader | swrite | nobody |
| Public | RWX | Invalid user | RWX | Invalid user |
| paraAdmin | R | Invalid user | RWX | Invalid user |
| lectura | RWX | RX | RX | Invalid user |

Archivo smb.conf

[global]

## Browsing/Identification ###

   netbios name= so-tpb

# Change this to the workgroup/NT-domain name your Samba server will part of

   workgroup = WORKGROUP

# server string is the equivalent of the NT Description field

   server string = %h - Sistema servidor

# This will prevent nmbd to search for NetBIOS names through DNS.

   dns proxy = no

wins support=yes

encrypt passwords = yes

#### Networking ####

interfaces = 192.168.1.0/24 eth0

#### Debugging/Accounting ####

# This tells Samba to use a separate log file for each machine

# that connects

   log file = /var/log/samba/log.%m

# Cap the size of the individual log files (in KiB).

   max log size = 1000

# If you want Samba to only log through syslog then set the following

# parameter to 'yes'.

#   syslog only = no

# We want Samba to log a minimum amount of information to syslog. Everything

# should go to /var/log/samba/log.{smbd,nmbd} instead. If you want to log

# through syslog you should set the following parameter to something higher.

   syslog = 0

# Do something sensible when Samba crashes: mail the admin a backtrace

   panic action = /usr/share/samba/panic-action %d

####### Authentication #######

# Server role. Defines in which mode Samba will operate. Possible

# values are "standalone server", "member server", "classic primary

# domain controller", "classic backup domain controller", "active

# directory domain controller".

#

# Most people will want "standalone sever" or "member server".

# Running as "active directory domain controller" will require first

# running "samba-tool domain provision" to wipe databases and create a

# new domain.

   server role = standalone server

# If you are using encrypted passwords, Samba will need to know what

# password database type you are using.

   passdb backend = tdbsam

   obey pam restrictions = yes

# This boolean parameter controls whether Samba attempts to sync the Unix

# password with the SMB password when the encrypted SMB password in the

# passdb is changed.

   unix password sync = yes

# For Unix password sync to work on a Debian GNU/Linux system, the following

# parameters must be set (thanks to Ian Kahan <<kahan@informatik.tu-muenchen.de> for

# sending the correct chat script for the passwd program in Debian Sarge).

   passwd program = /usr/bin/passwd %u

   passwd chat = \*Enter\snew\s\*\spassword:\* %n\n \*Retype\snew\s\*\spassword:\* %n\n \*password\supdated\ssuccessfully\* .

# This boolean controls whether PAM will be used for password changes

# when requested by an SMB client instead of the program listed in

# 'passwd program'. The default is 'no'.

   pam password change = yes

# This option controls how unsuccessful authentication attempts are mapped

# to anonymous connections

   map to guest = bad user

############ Misc ############

# Allow users who've been granted usershare privileges to create

# public shares, not just authenticated ones

   usershare allow guests = yes

#======================= Share Definitions =======================

[public]

   path=/samba/publico

   comment= carpeta publica a todos

   read only = no

   guest ok = yes

   browseable = yes

   invalid users= sreader

write list= @validos

[lectura]

   path=/samba/soloLectura

   comment= carpeta de solo lectura

   read only = yes

   guest ok = no

   browseable= yes

   valid users= @validos

   write list = @escribe

[paraAdmin]

   path=/samba/soloAutWrite

   comment=solo admin y root

   read only = no

   guest ok = no

   browseable= yes

   valid users= @admines, suser

   write list = @admines

Glosario

<https://www.samba.org/samba/docs/current/man-html/smb.conf.5.html>