Swinburne University of Technology

Unix for Telecommunications

Doubtfire Submission

Communications - Lab Report 3

Submitted By:
Nick Wiebenga
5378249
2020/10/14 22:35

Tutor:
Quoc Khanh Le

October 14, 2020

Produced by Doubtfire

Lab Report 3: Samba

Nick Wiebenga (5378249)
Swinburne University of Technology

***Abstract*—This lab report shows the commands and outputs/results of the Samba lab, and proceeds to describe how this setup is configured and tested. The Samba configuration file is created, transferred and edited according to the specification laid out in the lab handout. Required shares are created and tested for authentication and availability from a Windows PC.**

I. Aim

To make use of the Samba service to allow remote hosts to own, manage and access network shares.

II. Equipment

RULE host 42 (with Samba installed) via VPN connection from a home computer with PuTTY installed.

III. Method

(As per lab handout)

IV. Results

Below are listed the steps and their outcomes in sequence for this lab.

A. Locating smb4.conf

This file was copied before being edited.

```
root@rule42:/home/student # cd /usr/local/etc/
root@rule42:/usr/local/etc # ls *.conf
pkg.conf        smb4.conf       tcsd.conf
root@rule42:/usr/local/etc #
```

Fig. 1. Samba configuration file resides in /usr/local/etc

The Samba service was started using the command `service server_samba start`

B. Specifying the workgroup

The workgroup section is the first global setting in `smb4.conf`

```
# workgroup = NT-Domain-Name or Workgroup-Name, eg: MIDEARTH
   #workgroup = WORKGROUP
   workgroup = MSHOME
```

Fig. 2. The workgroup name is set to MSHOME

C. Setting the server string

The server string was edited

```
# server string is the equivalent of the NT Description field
   #server string = Samba Server
   server string = rule42.caia.swin.edu.au
```

Fig. 3. Server string set to the local host

D. Setting the Security Method

The security method is set to `users`

```
#security = user
security = user
```

Fig. 4. Security method set

E. Enabling Password Encryption

Lines added to smb4.conf within the 'security mode' section

```
   #security = user
   security = user
   encrypt passwords = yes
   smb password file = /usr/local/samba/private/smbpasswd
```

Fig. 5. Specifying encryption and location of the Samba password file

F. Allowing access from the Swinburne Network

The 'hosts allow' option determines which networks/hosts are allowed to connect

```
#following line SHOULD allow all Swinburne network
hosts allow = 136.186.
```

Fig. 6. All hosts within the 136.186.*.* range may connect

G. Enabling NTLMv1 authentication

NTLMv1 must be manually configured:

```
# Added line by myself for NTLMv1 authentication
   ntlm auth = yes
```

Fig. 7. This option is now necessary in Samba 4.5 and above

H. Creating Unix users

New users (samba and autocollector) were created
Input: `adduser`

```
root@rule42:/usr/local/etc # adduser
Username: samba
```

Fig. 8. Adding Unix users

I. Creating Samba users

Users must also have their corresponding Samba accounts
Input:
`smbpasswd -a <user>`

```
root@rule42:/usr/local/etc # smbpasswd -a samba
New SMB password:
```

Fig. 9. Adding Samba users

J. Creating shares for new users

Users require their own shares to be defined - this is done in `smb4.conf`, toward the end of the file:

```
# Lab definition 1 for 'samba'
[samba]
    comment = Samba
    path = /home/samba
    public = no
    writable = yes
    valid users = samba

# Lab definition 2 for 'autocollector'
[autocollector]
    comment = autocollector
    path = /home/autocollector
    public = no
    writable = yes
    valid users = autocollector
```

Fig. 10. Users samba and autocollector are given share definitions

K. Restarting the Samba service

Input: `service samba_server restart`

```
root@rule42:/usr/local/etc # service samba_server restart
Performing sanity check on Samba configuration: OK
Stopping smbd.
Waiting for PIDS: 17274, 17274.
Stopping nmbd.
Waiting for PIDS: 17268.
Performing sanity check on Samba configuration: OK
Starting nmbd.
Starting smbd.
root@rule42:/usr/local/etc #
```

Fig. 11. Samba is restarted for the configuration to take effect

L. Creating a 'share' subdirectory

```
root@rule42:/home # cd autocollector/
root@rule42:/home/autocollector # ls
.cshrc          .login          .login_conf
root@rule42:/home/autocollector # mkdir share
```

Fig. 12. A new directory created within autocollector's home directory

M. Viewing shares from Windows

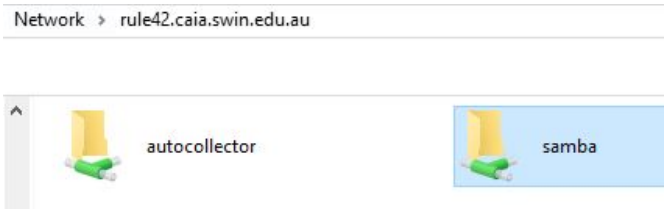

Fig. 13. Shares are visible from the Windows PC

N. Login attempt from Windows

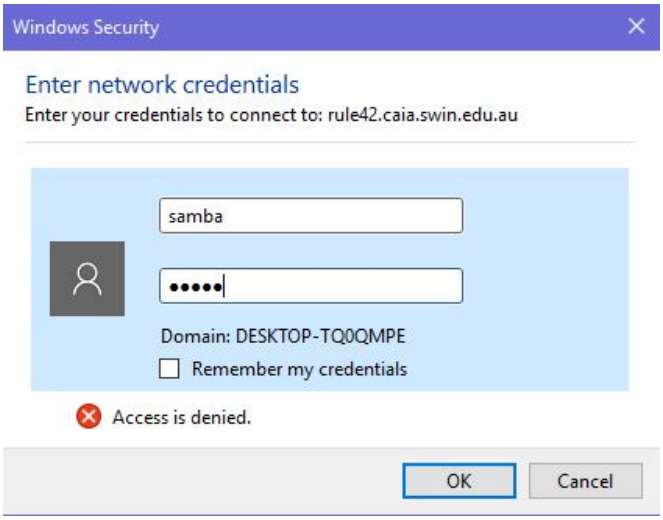

Fig. 14. Samba requests authentication

O. Viewing home directory from Windows

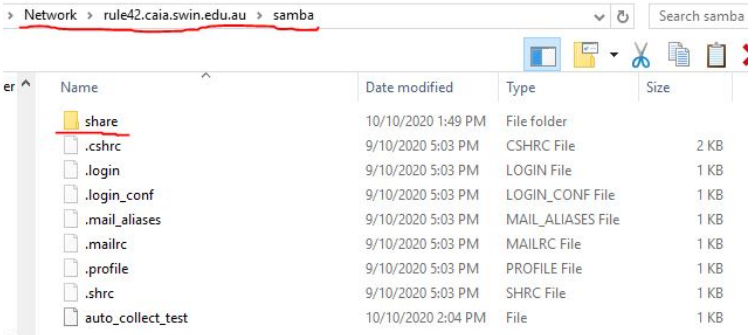

Fig. 15. The new 'share' subdirectory is not yet accessible

P. Share permissions

```
root@rule42:/home # ls -al
total 35
drwxr-xr-x  5 root          wheel          5 Oct  9 17:06 .
drwxr-xr-x  6 root          wheel         16 Aug  5 20:47 ..
drwxr-xr-x  2 autocollector autocollector  9 Oct  9 17:06 autocollector
drwxr-xr-x  2 samba         samba          9 Oct  9 17:03 samba
drwxr-xr-x  7 student       student       19 Oct  3 21:40 student
```

Fig. 16. Each user's home directory is accessible to them and self-owned

Q. Modifying 'share' subdirectory permission

```
root@rule42:/home/samba # ls -al
total 42
drwxr-xr-x  3 samba  samba    11 Oct 10 14:04 .
drwxr-xr-x  5 root   wheel     5 Oct  9 17:06 ..
-rw-r--r--  1 samba  samba  1054 Oct  9 17:03 .cshrc
-rw-r--r--  1 samba  samba   392 Oct  9 17:03 .login
-rw-r--r--  1 samba  samba   163 Oct  9 17:03 .login_conf
-rw-------  1 samba  samba   379 Oct  9 17:03 .mail_aliases
-rw-r--r--  1 samba  samba   339 Oct  9 17:03 .mailrc
-rw-r--r--  1 samba  samba   954 Oct  9 17:03 .profile
-rw-r--r--  1 samba  samba   851 Oct  9 17:03 .shrc
-rwxr--r--  1 samba  samba    52 Oct 10 14:04 auto_collect_test
drwxrwx---  2 root   samba     3 Oct 10 14:57 share
```

Fig. 17. The 'share' subdirectory was created by the root account

```
root@rule42:/home/samba # chown samba share
root@rule42:/home/samba # ls -al
total 42
drwxr-xr-x  3 samba  samba    11 Oct 10 14:04 .
drwxr-xr-x  5 root   wheel     5 Oct  9 17:06 ..
-rw-r--r--  1 samba  samba  1054 Oct  9 17:03 .cshrc
-rw-r--r--  1 samba  samba   392 Oct  9 17:03 .login
-rw-r--r--  1 samba  samba   163 Oct  9 17:03 .login_conf
-rw-------  1 samba  samba   379 Oct  9 17:03 .mail_aliases
-rw-r--r--  1 samba  samba   339 Oct  9 17:03 .mailrc
-rw-r--r--  1 samba  samba   954 Oct  9 17:03 .profile
-rw-r--r--  1 samba  samba   851 Oct  9 17:03 .shrc
-rwxr--r--  1 samba  samba    52 Oct 10 14:04 auto_collect_test
drwxrwx---  2 samba  samba     3 Oct 10 14:57 share
root@rule42:/home/samba #
```

Fig. 18. The 'share' subdirectory is now accessible to the account under whose home directory it resides

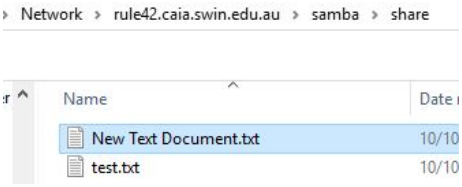

Fig. 19. File creation/modification is now possible as the 'samba' account

```
root@rule42:/usr/home # ls samba/share
New Text Document.txt   test.txt
root@rule42:/usr/home #
```

Fig. 20. File creation as seen from the Unix machine

R. Configuring Samba auto-start at boot

Input: `samba_server_enable="YES"`

```
#Auto-run Samba service at boot
samba_server_enable="YES"
```

Fig. 21. Changes made to /etc/rc.conf

V. Discussion

A. Ensuring Samba is installed

The Unix machine used for this lab had the Samba service already installed. If it were not installed, this can be done by issuing the command `pkg install samba4`. It can also be found in the ports directory `/usr/ports/net/samba/work/samba<version>`

B. Samba Configuration

The Samba service was started by issuing the command `service server_samba start`.
The primary means to configure Samba is the file `/usr/local/etc/smb4.conf`. This file does not initially exist on the RULE host, but a sample file was included as a lab resource. This file was transferred to the RULE host. Figure 1 shows this file in place.

1) Workgroup: In order for Samba to function, it must have information about the name of the workgroup it is serving; this is a collection of computers on a common LAN which share resources. Windows will have either `WORKGROUP` or `MSHOME` as the default workgroup. In Fig. 2 we have set the name to `MSHOME`.

2) Server string: The next configuration step is to set the server string. This is how the samba server identifies itself to other machines, and provides a way to connect via a name (as will be seen later on the Windows PC). Fig. 3 shows this configured as `rule42.caia.swin.edu.au`, but it could be given any other meaningful name.

3) Security Method + Encryption: There are a number of security methods available to Samba, including a legacy method called 'Server Security'. Others are ADS (for native Active Directory support), Domain Security Mode, Share-Level Security, and User Level Security[1]. This is a simpler method based not on specific resources (as would be the case in Share-Level) but rather user authentication and whether the remote machine name is permitted. Fig. 4 shows the line required in `smb4.conf` for User Level security.

We also desire this authentication to be encrypted so as to avoid potential packet-sniffing by the likes of TCPDump, and in Figure 5 the lines specify not only that passwords be encrypted but where to store them locally (as is required for this security method).

4) Permitting network access: Since this lab is to be marked by a machine on the local network as well as accessed by a machine (via VPN) within the Swinburne 136.186.*.* subnet, access needs to be permitted accordingly. This is accomplished by the line seen in Fig. 6. The less specific this line is, the greater the number of IP addresses allowed access.

Exceptions can also be made. For example, one may wish to allow the entire aforementioned subnet aside from RULE43. This would be achieved with the line '`hosts allow = 136.186. except 136.186.230.43`'. This section of `smb4.conf` is effectively an IP-level form of access control.

5) Enabling NTLMv1: Beginning with Samba version 4.5, NTLM version 1 is disabled by default, requiring NTLMv2 which not all clients support. More importantly, MSCHAPv2 over VPNs make use of NTLMv1, which will be necessary for this lab given the equipment and connection being used. Again, this configuration is entered in the `smb4.conf` file, as shown in Fig. 7. The default is either `ntlm auth = no` or no line at all, as was the case in this lab.

C. Creating Unix and Samba users

Samba has its own database of users which will be created. Before doing this however, they will be created as regular Unix users:

- Username: samba, Password: samba
- Username: autocollector, Password: autocollector

The command `adduser` is used, after which a series of prompts appear relating to the username, password, home directory, default shell and other properties. The command and first prompt can be seen in Fig. 8 for the user 'samba'. Default values were used where unspecified in the lab handout.

Creating the same two users for Samba requires the '`smbpasswd -a`' command, followed by the username to be added.
Other options are available, such as removing users or simply disabling them in the local `smbpasswd` file[2]. Fig. 9 shows the user 'samba' being added using this tool.

D. Defining shares

Network shares require definition so that the Samba server is aware of their location, visibility, and access permissions. Fig. 10 shows the two share definitions created for each of the new Samba users created previously.

We will examine the `samba` share only, as both shares were given the same rules.

The first line is the Samba server name which the clients will see, in this case `samba`.

The second line, `comment`, is simply a name to associate with the share, similar to how an alias operates in DNS configurations; it can be a more human-friendly descriptor such as "The Samba Share".

The next line defines the path to the shared folder. In keeping with the established structure of `/usr/home/<user>`, share paths are configured such that they line up with existing home directories.

The fourth line mandates that guest accounts are not permitted access to the share (synonymous with `guest ok = no`). This ensures that authentication is required, because only a user account can access the share.

Line five permits write access for given valid users to the share. The default behaviour for Samba is that accessible shares are read-only, so for our users to be able to do more than read, this option needs to be set as shown.

The final line specifies the users permitted to access the share. In the example shown, only one user per share is allowed, however it is possible to have a list of valid users (separated by commas) and a list of invalid users (using the line `invalid users = <user1>, <user2>...`

E. Resarting the Samba service

For the configuration changes to take effect, the service needs to be restarted. This is shown in Fig. 11.

F. Creating a subdirectory

The lab did not actually call for this action, but as an extra step to test functionality it was deemed useful from an academic standpoint. The command (seen in Fig. 12) to perform this is `mkdir <dir>` .

G. Connecting from Windows PC

The address `\\rule42.caia.swin.edu.au` was entered in Windows Explorer's navigation bar. The result of this is shown in Fig. 13 where the visible shares are displayed.

H. Browsing to the 'samba' share

Attempting to view the contents of the 'samba' share evoked an authentication prompt by Windows Security as seen in Fig. 14. After entering the username and password, this share can be viewed (Fig. 15). The newly created subdirectory is also seen here. After this has been done once, it will not be asked again as the credentials are cached - this can be cleared via the command: '`net use \\rule42.caia.swin.edu.au /delete`' in Windows Powershell.

I. File-level Permissions

The subdirectory 'share' seen in Fig. 15 was inaccessible to user samba. This is because it was created by the root user. During the account creation process, the home directories were manually assigned to each user with them as owner, as seen in Figure 16, however when the new subdirectory was created within user samba's home directory, the default owner is the creator (root) shown in Figure 17. This was modified with the command chown as seen in Fig. 18 which also shows the result of the change of ownership.

J. Testing the change to permissions

From the Windows PC as user samba, new test text files were successfully created (Fig. 19), and this change is reflected when viewed from the Unix machine (Fig. 20).

K. Configuring Samba to start at bootup

As with previous labs dealing with service configuration, in order for the service to be started as part of the boot procedure, the file /etc/rc.conf has a line appended. This line is samba_server_enable="YES", seen in Figure 21.

VI. Conclusion:

This lab explored Samba server configuration. Samba allows for resource sharing between different operating systems such as Linux and Microsoft Windows via the SMB protocol[3]. Samba also can allow for Active Directory roles such as a Domain Controller or member. This means not only a common resource sharing protocol is necessary, but the ability to provide cross-platform authorisation and authentication, as well as name resolution. This flexibility enables administrators to oversee mixed environments with interoperability.

VII. References

[1] - Tridgell A et al 'Server Types and Security Modes' viewed 11th October 2020, https://www.samba.org/samba/docs/old/Samba3-HOWTO/ServerType.html#id2559114

[2] - FreeBSD smbpasswd man page 'SMBPASSWD(8)' viewed 12th October 2020, https://www.freebsd.org/cgi/man.cgi?query=smbpasswd&format=html

[3] - FreeBSD Samba man page 'SAMBA(8)' viewed 12th October 2020, https://www.freebsd.org/cgi/man.cgi?samba