Networking Lab Assignment 18

Network with multiple subnets with wired and wireless LANs

Albin Antony

16 May 2019

1 Process And Thread

1.1 Aim

Design and configure a network with multiple subnets with wired and wireless LANs using required network devices. Configure the following services in the network - TELNET, SSH, FTP server, Web server, File server, DHCP server and DNS server.

1.2 Theory

1.2.1 Subnet

A subnet is a logical partition of an IP network into multiple, smaller network segments. It is typically used to subdivide large networks into smaller, more efficient subnetworks. The internet is composed of many networks that are run by many organizations. In turn, each organization's network can be composed of many smaller networks, or subnets. Each subnet allows its connected devices to communicate with each other, and routers are used to communicate between subnets. The size of a subnet depends on the connectivity requirements and the network technology employed. A point-to-point subnet allows two devices to connect, while a data center subnet might be designed to connect many more devices.

1.3 Configuring the Services

The following shows how the different services can be configured in an Ubuntu PC:

1.3.1 Telnet

Telnet is a user command and an underlying TCP/IP protocol for accessing remote computers. Through Telnet, an administrator or another user can access someone else's computer remotely. On the Web, HTTP and FTP protocols allow you to request specific files from remote computers, but not to actually be logged on as a user of that computer. With Telnet, you log on as a regular user with whatever privileges you may have been granted to the specific application and data on that computer. Telnet is most likely to be used by program developers and anyone who has a need to use specific applications or data located at a particular host computer.
Take the following steps to configure Telnet:

- Install Telnet

```
1        sudo apt install telnet xinetd
```

- Edit /etc/inetd.conf with root permission,add this line:

```
        telnet stream tcp nowait telnetd /usr/sbin/tcpd
            /usr/sbin/in.telnetd
```

- Edit /etc/xinetd.conf, copy the following configuration:

```
# Simple configuration file for xinetd
#
# Some defaults, and include /etc/xinetd.d/
defaults
{
# Please note that you need a log_type line to be able
    to use log_on_success
# and log_on_failure. The default is the following :
# log_type = SYSLOG daemon info
instances = 60
log_type = SYSLOG authpriv
log_on_success = HOST PID
log_on_failure = HOST
cps = 25 30
}
```

- Change telnet port by using the following command in the terminal:

```
        telnet 23/tcp
```

- Then restart the service:

```
        sudo /etc/init.d/xinetd restart
```

1.3.2 SSH

The SSH protocol (also referred to as Secure Shell) is a method for secure remote login from one computer to another. It provides several alternative options for strong authentication, and it protects the communications security and integrity with strong encryption. It is a secure alternative to the non-protected login protocols (such as telnet, rlogin) and insecure file transfer methods (such as FTP).
Take the following steps to configure SSH:

- Install SSH:

```
        sudo apt-get install openssh-server
```

  (Installing the client can be done by replacing openssh-server by openssh-client)

- Configure SSH:

```
        sudo nano /etc/ssh/sshd_config
```

Then make the changes you want to make.

- Restart SSH:

```
sudo systemctl restart ssh
```

We can login to the SSH server from an SSH client.

1.3.3 FTP Server

File Transfer Protocol (FTP) is the commonly used protocol for exchanging files over the Internet. FTP uses the Internet's TCP/IP protocols to enable data transfer. FTP uses a client-server architecture, often secured with SSL/TLS. FTP promotes sharing of files via remote computers with reliable and efficient data transfer. FTP uses a client-server architecture. Users provide authentication using a sign-in protocol, usually a username and password, however some FTP servers may be configured to accept anonymous FTP logins where you don't need to identify yourself before accessing files. Most often, FTP is secured with SSL/TLS.
The following steps show setting up an FTP server on the computer:

- Install FTP daemon:

```
sudo apt install vsftpd
```

- Configuring FTP can be done by editing the following file:

```
/etc/vsftpd.conf
```

- Restart the service:

```
sudo systemctl restart vsftpd.service
```

1.3.4 Web Server

A Web server is a program that uses HTTP (Hypertext Transfer Protocol) to serve the files that form Web pages to users, in response to their requests, which are forwarded by their computers' HTTP clients. Dedicated computers and appliances may be referred to as Web servers as well. The process is an example of the client/server model. All computers that host Web sites must have Web server programs. Leading Web servers include Apache (the most widely-installed Web server), Microsoft's Internet Information Server (IIS) and nginx (pronounced engine X) from NGNIX. Other Web servers include Novell's NetWare server, Google Web Server (GWS) and IBM's family of Domino servers.
A web server can be hosted on the localhost of the PC by following the following steps:

- Installing the server: The most common server on Linux systems and it is called the LAMP server. It can be installed on Ubuntu by:

```
sudo apt install lamp-server^
```

- Hosting a website: By creating a *.conf* file in the */etc/apache2/sites-available/* folder, we inform the server of the location of the code for our website.
- Enabling the website by using the command:

```
sudo a2ensite <nameOfFile.conf>
```

- By editing */etc/hosts* file, we can give the domain name for the website
- The configuration the server is in the file: */etc/apache2/apache2.conf*
- Restart the server by the command:

```
sudo systemctl restart apache2.service
```

1.3.5 File Server

In the client/server model, a file server is a computer responsible for the central storage and management of data files so that other computers on the same network can access the files. A file server allows users to share information over a network without having to physically transfer files by floppy diskette or some other external storage device. Any computer can be configured to be a host and act as a file server. In its simplest form, a file server may be an ordinary PC that handles requests for files and sends them over the network. In a more sophisticated network, a file server might be a dedicated network-attached storage (NAS) device that also serves as a remote hard disk drive for other computers, allowing anyone on the network to store files on it as if to their own hard drive.
The following steps can be followed to setup a file server:

- Installing Samba File Server:

```
sudo apt install samba
```

- Configuring the file server by editing */etc/samba/smb.conf*

  First, edit the following key/value pairs in the [global] section of /etc/samba/smb.conf:

```
workgroup = EXAMPLE
...
security = user
```

  Create a new section at the bottom of the file, or uncomment one of the examples, for the directory to be shared:

```
[share]
 comment = Ubuntu File Server Share
 path = /srv/samba/share
 browsable = yes
 guest ok = yes
 read only = no
 create mask = 0755
```

- Make a directory for hosting files and setting permission for the directory:

```
sudo mkdir -p /srv/samba/share
sudo chown nobody:nogroup /srv/samba/share/
```

- Restart Samba service:

```
sudo systemctl restart smbd.service nmbd.service
```

1.3.6 DHCP Server

DHCP (Dynamic Host Configuration Protocol) is a network management protocol used to dynamically assign an Internet Protocol (IP) address to any device, or node, on a network so they can communicate using IP. DHCP automates and centrally manages these configurations rather than requiring network administrators to manually assign IP addresses to all network devices. DHCP can be implemented on small local networks as well as large enterprise networks. DHCP will assign new IP addresses in each location when devices are moved from place to place, which means network administrators do not have to manually initially configure each device with a valid IP address or reconfigure the device with a new IP address if it moves to a new location on the network. Versions of DHCP are available for use in Internet Protocol version 4 (IPv4) and Internet Protocol version 6 (IPv6).
The following steps shows how DHCP server can be run:

- Install DHCP server:

```
sudo apt-get install isc-dhcp-server
```

- Configure DHCP server, the config file is */etc/dhcp/dhcpd.conf*:

```
# Sample /etc/dhcpd.conf
# (add your comments here)
default-lease-time 600;
max-lease-time 7200;
option subnet-mask 255.255.255.0;
option broadcast-address 192.168.1.255;
option routers 192.168.1.254;
option domain-name-servers 192.168.1.1, 192.168.1.2;
option domain-name "mydomain.example";
```

```
10
11 subnet 192.168.1.0 netmask 255.255.255.0 {
12 range 192.168.1.10 192.168.1.100;
13 range 192.168.1.150 192.168.1.200;
14 }
```

- Starting and stopping services can be achieved using:

```
1             sudo service isc-dhcp-server restart
2             sudo service isc-dhcp-server start
3             sudo service isc-dhcp-server stop
```

  After editing configuration files, we have to restart the service.

1.3.7 DNS Server

The Domain Name Systems (DNS) is the phonebook of the Internet. Humans access information online through domain names, like nytimes.com or espn.com. Web browsers interact through Internet Protocol (IP) addresses. DN S translates domain names to IP addresses so browsers can load Internet resources.

Each device connected to the Internet has a unique IP address which other machines use to find the device. DNS servers eliminate the need for humans to memorize IP addresses such as 192.168.1.1 (in IPv4), or more complex newer alphanumeric IP addresses such as 2400:cb00:2048:1::c629:d7a2 (in IPv6).
The following steps show the setup:

- Installing:

```
1             sudo apt install bind9
```

- The configuration is in the */etc/bind* folder
- Setting as a catching name server by
  editing the file */etc/bind/named.conf.options*:

```
1 forwarders {
2 1.2.3.4; # replace with the ip address
3 5.6.7.8; # of the name servers
4 };
```

- BIND9 can be configured with the primary and the secondary master as a custom DNS server to access all the subnets.
- Restarting bind9:

```
1             sudo systemctl restart bind9.service
```

1.4 Result

For accessing the different nodes in the subnet, TELNET, SSH, FTP server, Web server, File server, DHCP server and DNS server have been configured and runs successfully in an Ubuntu 16.04 PC.