SSBT's College of Engineering & Technology, Bambhori, Jalgaon
Department of Computer Applications

Practical: 04

DOP: **DOC:**

Title: Installation, Configuration and Administration of Internet Servers
1)SSH Server
2)FTP Server
3)File Server (Samba)
4)HTTP Server

Objective : Implement system administration tasks, installation, configuration and administration of SSH server, FTP server , file server ,HTTP server and internet servers.

4.1 Aim: - Installation and Configuration SSH Server (Ubuntu enable/disable root login.)

Objectives:
1. Students should able to install SSH server and configure it on Computer System and even on virtual environment
2. Students should able to Differentiate between secure and unsecure remote system access.

THEORY:

sshd (OpenSSH Daemon or server) is the daemon program for ssh client. It is a free and open source ssh server. ssh replaces insecure rlogin and rsh, and provide secure encrypted communications between two untrusted hosts over an insecure network such as the Internet. Ubuntu Desktop and minimal Ubuntu server do not come with sshd installed.

Installation steps to be followed: (Ubuntu)
1. Open the terminal application for Ubuntu desktop.
2. Type sudo apt-get install openssh-server
3. Enable the ssh service by typing sudo systemctl enable ssh
4. Start the ssh service by typing sudo systemctl start ssh
5. Test it by login into the system using ssh user@server-name

CONCLUSIONS:
In this Practical we learned how to enable SSH on a Ubuntu Additionally, we configured firewall and SSH rules to limit access.

4.2 Aim: - Installation and Configuration of FTP Server (Ubuntu).

THEORY:
FTP (File Transfer Protocol) is a relatively old and most used standard network protocol used for uploading/downloading files between two computers over a network.
However, FTP by its original insecure, because it transmits data together with user credentials (username and password) without encryption. FTP is unencrypted by default, so by itself; it is not a good choice for secure transmission of data

Installation steps to be followed: (Ubuntu)

Step 1: Update System Packages

Start by updating your repositories – enter the following in a terminal window:

```
sudo apt-get update
```

The system proceeds to update the repositories.

Step 2: Backup Configuration Files

Before making any changes, make sure to back up your configuration files.
1. Create a backup copy of the default configuration file by entering the following:

```
sudo cp /etc/vsftpd.conf /etc/vsftpd.conf_default
```

This command creates a copy of the default configuration file.
2. Create a new vsftpd configuration file /etc/vsftpd.conf using your preferred text editor:

```
$ sudo gedit /etc/vsftpd.conf
```

Step 3: Install vsftpd Server on Ubuntu

A common open-source FTP utility used in Ubuntu is vsftpd. It is recommended for its ease of use.
1. To install vsftpd, enter the command:

```
sudo apt install vsftpd
```

2. To launch the service and enable it at startup:

```
sudo systemctl start vsftpd
sudo systemctl enable vsftpd
```

Step 4: Create FTP User

Create a new FTP user with the following commands:

```
sudo useradd –m testuser
sudo password testuser
```

The system should ask you to create a password for the new testuser account. Create a sample file in the new user's home account:

```
sudo mkdir /home/testuser
```

Step 5: Configure Firewall to Allow FTP Traffic

If you are using UFW that comes standard with Ubuntu, it will block FTP traffic by default. Enter the following commands to open Ports 20 and 21 for FTP traffic:

```
sudo ufw allow 20/tcp
sudo ufw allow 21/tcp
```

Step 6: Connect to Ubuntu FTP Server

Connect to the FTP server with the following command:

```
sudo ftp ubuntu-ftp
```

Replace ubuntu-ftp with the name of your system (taken from the command line).
Log in using the testuser account and password you just set. You should now be successfully logged in to your FTP server.

CONCLUSIONS:
In this Practical we learned how to install and enable FTP Server on a Ubuntu .
By following above steps you should have installed an FTP server on Ubuntu with **vsftpd.**
You should now be able to configure your user lists and accounts, and connect to your new FTP server.

4.3 **Aim: - Installation and Configuration of Samba Server (Ubuntu).**

THEORY:
Samba is a free and open-source re-implementation of the SMB/CIFS network file sharing protocol that allows end users to access files, printers, and other shared resources.
A Samba file server enables file sharing across different operating systems over a network. It lets you access your desktop files from a laptop and share files with Windows and macOS users.
Samba has both client and server components. Samba uses the SMB protocol, which is necessary when accessing assets on a file server from a Microsoft computer. Samba can also work as a domain controller that is compatible with Microsoft Active Directory

Installation steps to be followed: (Ubuntu)

Step 1: To install Samba, we run:
sudo apt update
sudo apt install samba

Step 2: Setting up Samba
Now that Samba is installed, we need to create a directory for it to share:
mkdir /home/<username>/sambashare/
The command above creates a new folder sambashare in our home directory which we will share later.
The configuration file for Samba is located at /etc/samba/smb.conf. To add the new directory as a share, we edit the file by running:

sudo nano /etc/samba/smb.conf

At the bottom of the file, add the following lines:
[sambashare]
comment = Samba on Ubuntu
path = /home/username/sambashare
read only = no
browsable = yes
Then press Ctrl-O to save and Ctrl-X to exit from the nano text editor.
Now that we have our new share configured, save it and restart Samba for it to take effect:
sudo service smbd restart

Step 3: Update the firewall rules to allow Samba traffic:
sudo ufw allow samba

Step 4: Setting up User Accounts and Connecting to Share
Since Samba doesn't use the system account password, we need to set up a Samba password

for our user account:

```
sudo smbpasswd -a username
```

Step 5: Connecting to a Samba Share from Windows

Windows users also have an option to connect to the Samba share from both command line and GUI. The steps below show how to access the share using the Windows File Explorer.

1. Open up File Explorer and in the left pane right-click on “This PC”.
2. Select “Choose a custom network location” and then click “Next”.
3. In “Internet or network address”, enter the address of the Samba share in the following format \\samba_hostname_or_server_ip\sharename

Conclusion:

In this Practical, you have learned how to install a Samba server on Ubuntu and create different types of shared and users. We have also learn how to connect to the Samba server from Linux and Windows devices

4.4 **Aim: - Installation and Configuration of HTTP Server (Ubuntu).**

THEORY:

The Apache HTTP server is the most widely-used web server in the world. It provides many powerful features, including dynamically loadable modules, robust media support, and extensive integration with other popular software.

The Apache HTTP Server Project is an effort to develop and maintain an open-source HTTP server for modern operating systems including UNIX and Windows. The goal of this project is to provide a secure, efficient and extensible server that provides HTTP services in sync with the current HTTP standards.

The Apache HTTP Server ("httpd") was launched in 1995 and it has been the most popular web server on the Internet since April 1996. It has celebrated its 25th birthday as a project in February 2020.

Installation steps to be followed: (Ubuntu)

Step 1: Installing Apache

Apache is available within Ubuntu’s default software repositories, so you can install it using conventional package management tools.

Update your local package index:

```
sudo apt update
```

Install the apache2 package:

```
sudo apt install apache2
```

Step 2: Configuration of firewall

```
sudo ufw allow 'Apache
```

Step 3: Checking Web Server

Check with the systemd init system to make sure the service is running by typing:

```
sudo systemctl status apache2
```

Access the default Apache landing page to confirm that the software is running properly through your IP address:

```
http://your_server_ip
```

Conclusion:

In this Practical, we have installed and managed the Apache web server.