

Birla Institute of Technology & Science, Pilani
Computer Networks (CS F 303 / IS F 303)
Second Semester 2015-2016

Lab Session: 4

Aim: To learn configuring an FTP server in Linux environment and analysis of FTP protocol using Wireshark.

Objective: Our objective is to learn to setup a File Transfer Protocol (FTP) server to host our own data for access from anywhere. Other objective is to analyse FTP using Wirshark, one can look into flow graph (through Wireshark) to understand the working of FTP. Also, many more features can be observed such as protocol hierarchy etc.

Description: The File Transfer Protocol (FTP) is used as one of the most common means of copying files between servers over the Internet. It is a standard network protocol operating on the application layer of the OSI model and is used to transfer files using TCP/IP. Most web based download sites use the built in FTP capabilities of web browsers and therefore most server oriented operating systems usually include an FTP server application as part of the software suite. Linux is no exception. This lab sheet will show you how to convert your Linux box into an FTP server using Very Secure FTP Daemon (VSFTPD) package.

It operates in two connection channels:

- ✧ **FTP Control Channel, TCP Port 21:** All commands you send and the ftp server's responses to those commands will go over the control connection
- ✧ **FTP Data Channel, TCP Port 20:** This port is used for all subsequent data transfers between the client and server.

From a networking perspective, the two main types of FTP are active and passive. In active FTP, the FTP server initiates a data transfer connection back to the client. For passive FTP, the connection is initiated from the FTP client.

For more information, jump on to <http://www.w3.org/Protocols/rfc959/>

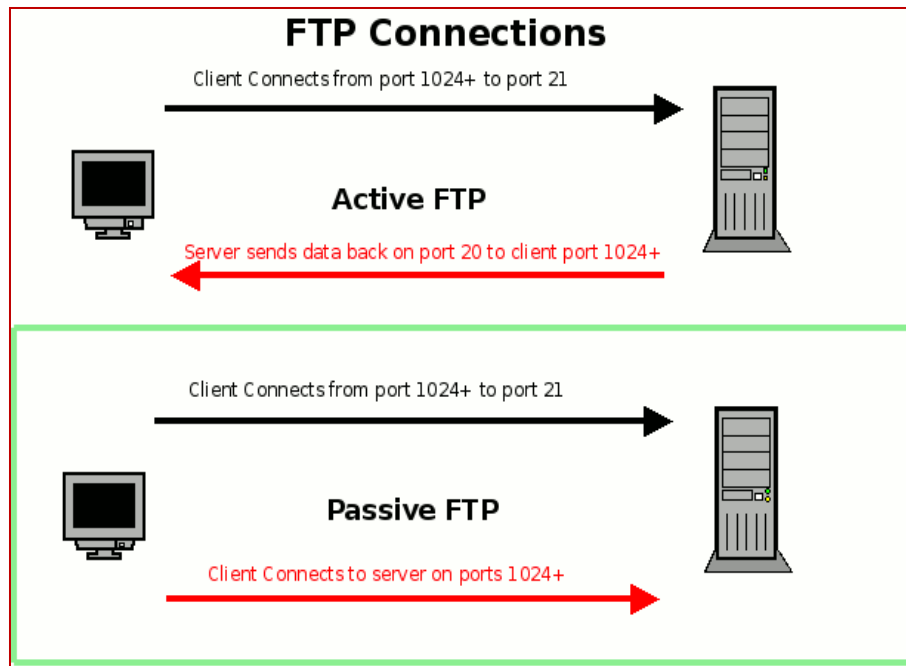


Figure 1

PART-1: Configuration of FTP Server (VSFTPD)

Installing VSFTPD:

vsftp can be easily deployed on your machine with the following simple code from the command-line terminal:

➤ *\$sudo apt-get install vsftpd*

Alternatively, you can get it by searching for vsftpd in Ubuntu Software Centre.

Note: To avoid overburdening the network, download VSFTPD through BITSFOSS apt-catcher service. To know more about configuring your proxy for Ubuntu in such a way, visit <http://bitsfoss.bits-pilani.ac.in/home/services/ubuntu-users/>

Procedure:

Step:1- Starting vsftpd:

You can use the `sysv-rc-conf` command to get VSFTPD configured to start at boot:

➤ *\$sudo sysv-rc-conf vsftpd on*

To start, stop, and restart VSFTPD after booting by:

- `$sudo service vsftpd start`
- `$sudo service vsftpd stop`
- `$sudo service vsftpd restart`

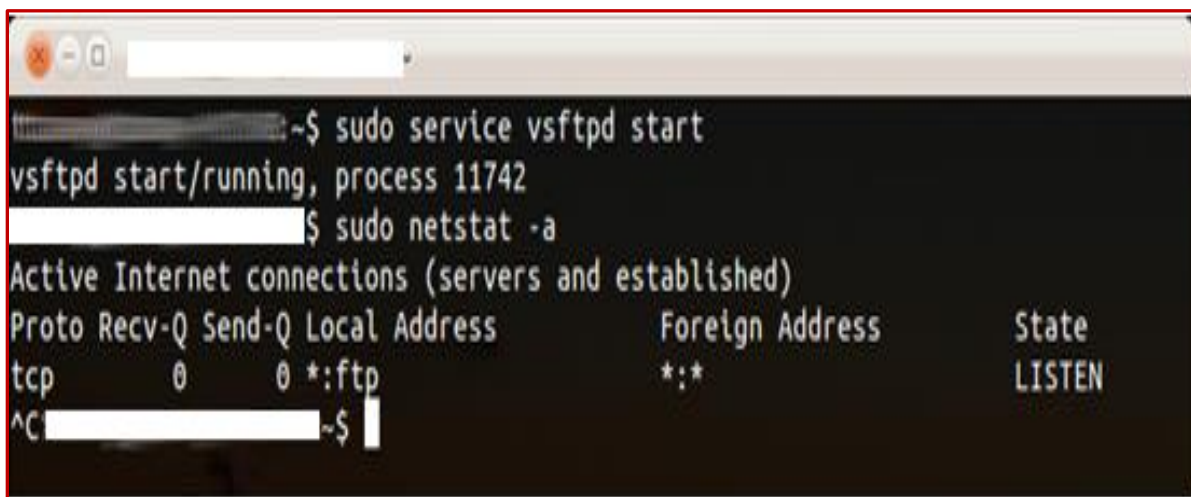
To determine whether VSFTPD is running you can issue either of these two commands. The first will give a status message. The second will return the process ID numbers of the VSFTPD daemons.

- `$sudo service vsftpd status`
- `$pgrep vsftpd`

To test the status of VSFTPD and list all the TCP and UDP ports on which the server is listening for traffic:

- `$sudo netstat -a | grep ftp`

If vsftpd wasn't running, there would be no such output at all.

A terminal window screenshot showing the execution of two commands. The first command is `sudo service vsftpd start`, which returns `vsftpd start/running, process 11742`. The second command is `sudo netstat -a`, which displays a table of active internet connections. The table has columns for Protocol, Receive Queue, Send Queue, Local Address, Foreign Address, and State. The output shows a single entry for TCP on port ftp (21) in a LISTEN state.

```
~$ sudo service vsftpd start
vsftpd start/running, process 11742
~$ sudo netstat -a
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 *:ftp                    *:*                     LISTEN
^C ~$
```

Figure 2

Step:2- Configuring vsftpd.conf file:

The file may be located in the `/etc` folder in Ubuntu 11.10. Get to the file browser (nautilus) and open `/etc`. Next scroll down and double click on a file called `vsftpd.conf`.

Remember that lines that start with a '#' are commented out. Ensure the following:

1. Disable anonymous access:

VSFTPD runs as an anonymous FTP server. Anonymous FTP is the choice of Web sites that need to exchange files with numerous unknown remote users. Unlike regular FTP where you login with a preconfigured Linux username and password, anonymous FTP requires only a username of anonymous and your email address for the password.

✓ *Change the “anonymous_enable” setting to NO.*

2. Change the "local_enable" setting:

You'll also need to simultaneously enable local users to be able to log in by removing the comment symbol (#) before the local_enable instruction.

✓ *Change the “local_enable” to YES.*

3. Restrict user access to FTP directory only:

You may restrict local users to their home directories. You should uncomment this option keeping in mind security issues with open access to your other folders.

✓ *Change the “chroot_local_user” to YES.*

4. Optional (for anonymous FTP access only):

If you enable anonymous FTP with VSFTPD, remember to define the root directory that visitors will visit. This is done with the anon_root directive:

anon_root=/data/directory

⚠ VSFTPD allows only anonymous FTP downloads to remote users, not uploads from them. This can be changed by modifying the anon_upload_enable directive.

^ VSFTPD doesn't allow anonymous users to create directories on your FTP server. You can change this by modifying the `anon_mkdir_write_enable` directive.

^ VSFTPD logs FTP access to the `/var/log/vsftpd.log` log file. You can change this by modifying the `xferlog_file` directive.

^ By default VSFTPD expects files for anonymous FTP to be placed in the `/var/ftp` directory. You can change this by modifying the `anon_root` directive.

There are many other options you can add to this file like limiting the maximum number of client connections (`max_clients`), maximum rate of data transfer per non-anonymous login (`local_max_rate`). Descriptions on this and more can be found in the `vsftpd.conf` man pages:

➤ *\$sudo man vsftpd.conf*

VSFTPD only reads the contents of its `vsftpd.conf` configuration file only when it starts, so you'll have to restart VSFTPD each time you edit the file in order for the changes to take effect.

Tutorial #1:

In this example, anonymous FTP is not desired, but a group of trusted users need to have read only access to a directory for downloading files.

1. Start a terminal and login as root for furth.

➤ *\$sudo -i*

2. Start VSFTP.

3. Create a user group and shared directory. In this case, use `/home/ftp-docs` and a user group name of `ftp-users` for the remote users:

➤ *#groupadd ftp-users*

➤ *#mkdir /home/ftp-docs*

4. Make the directory accessible to the `ftp-users` group:

➤ *#chmod 750 /home/ftp-docs*

750 means 7 (rwx) for the owner, 5 (r-x) for the group and 0 (- - -) for others:

➤ *#chown root:ftp-users /home/ftp-docs*

5. Add users, and make their default directory /home/ftp-docs:

- *#useradd -g ftp-users -d /home/ftp-docs user1*
- *#useradd -g ftp-users -d /home/ftp-docs user2*
- *#passwd user1*

You'll be prompted to enter and then retype a new UNIX password for the user.

- *#passwd user2*

6. Copy files to be downloaded by your users into the /home/ftp-docs directory through the file explorer.

- *#nautilus /home/ftp-docs*

7. Change the permissions of the files in the /home/ftp-docs directory for read only access by the group:

- *#chown root:ftp-users /home/ftp-docs/**
- *#chmod 740 /home/ftp-docs/**

8. Log off as the root when you are done configuring the FTP:

- *#logout*

Users should now be able to log in via FTP to the server using their new usernames and passwords. If you absolutely don't want any FTP users to be able to write to any directory, then you should change the `write_enable` line in your `vsftpd.conf` file:

write_enable = NO

Remember, you must restart VSFTPD for the configuration file changes to take effect.

Sample login session:

To learn about more access commands for FTP, go through the documentation provided with the default FTP client on your machine by:

- *\$man ftp*

An FTP server can also be accessed in a user interface manner through a basic web browser by supplying *ftp://<ip address of the FTP server>* in the address bar.

```

tanay@tanay-M17xR3:~$ sudo -i
[sudo] password for tanay:
root@tanay-M17xR3:~# ftp 172.17.1.41
Connected to 172.17.1.41.
220 Welcome to this tutorial's FTP service.
Name (172.17.1.41:tanay): user2
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
-rw-rw-r-- 1 0 0 236121 Mar 01 21:18 Configuration of FTP server.odt
-rw-rw-r-- 1 1000 1000 28 Mar 01 21:38 random.txt
-rw-rw-r-- 1 1000 1000 23 Mar 01 20:54 random.txt~
226 Directory send OK.
ftp> get random.txt
local: random.txt remote: random.txt
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for random.txt (28 bytes).
226 Transfer complete.
28 bytes received in 0.00 secs (201.1 kB/s)
ftp> quit
221 Goodbye.
root@tanay-M17xR3:~# logout
tanay@tanay-M17xR3:~$

```

Figure 3

PART-2: Analysis of FTP Protocol

FTP - Packet Sniffing and Wireshark Analysis

FTP (File Transfer Protocol) as the name implies, is a protocol to transfer files from one computer to another. The protocol operates on TCP ports 20 for data transfer and 21 for control. The authentication, communication, and file data are all communicated in plain text; meaning *no* encryption is used. The most common implementation of FTP is establishing a connection and transferring files with a web server.

You can use currently installed FTP server for the analysis of FTP protocol. You can find out the set of protocols that are workhorse for FTP protocol. Also you can analyze various network

parameters related to FTP. The statistic option of Wirshark allows you to analyze various graphs. Let us go through all this packet capturing and analysis of FTP using Wireshark.

Here are the steps for capturing and analysing FTP Protocol:

Step: 1 Start a Wireshark capture

- a. Close all unnecessary network traffic, such as the web browser, to limit the amount traffic during the Wireshark capture.
- b. Start the Wireshark capture.

Step: 2 Download Files

- a. Download some file using your currently configured FTP server.

OR

- a. From the command prompt, enter *ftp ftp.cdc.gov*
- b. Log into the FTP site for Centres for Disease Control and Prevention (CDC) with user anonymous and no password.
- c. Locate and download the Readme file.

Note: See Figure No-4 and Figure No-5.



```
C:\>ftp ftp.cdc.gov
Connected to ftp.cdc.gov.
220 Microsoft FTP Service
User (ftp.cdc.gov:(none)): anonymous
331 Anonymous access allowed, send identity (e-mail name) as password.
Password:
230 User logged in.
ftp> get readme
200 PORT command successful.
150 Opening ASCII mode data connection.
-
```

Figure 4

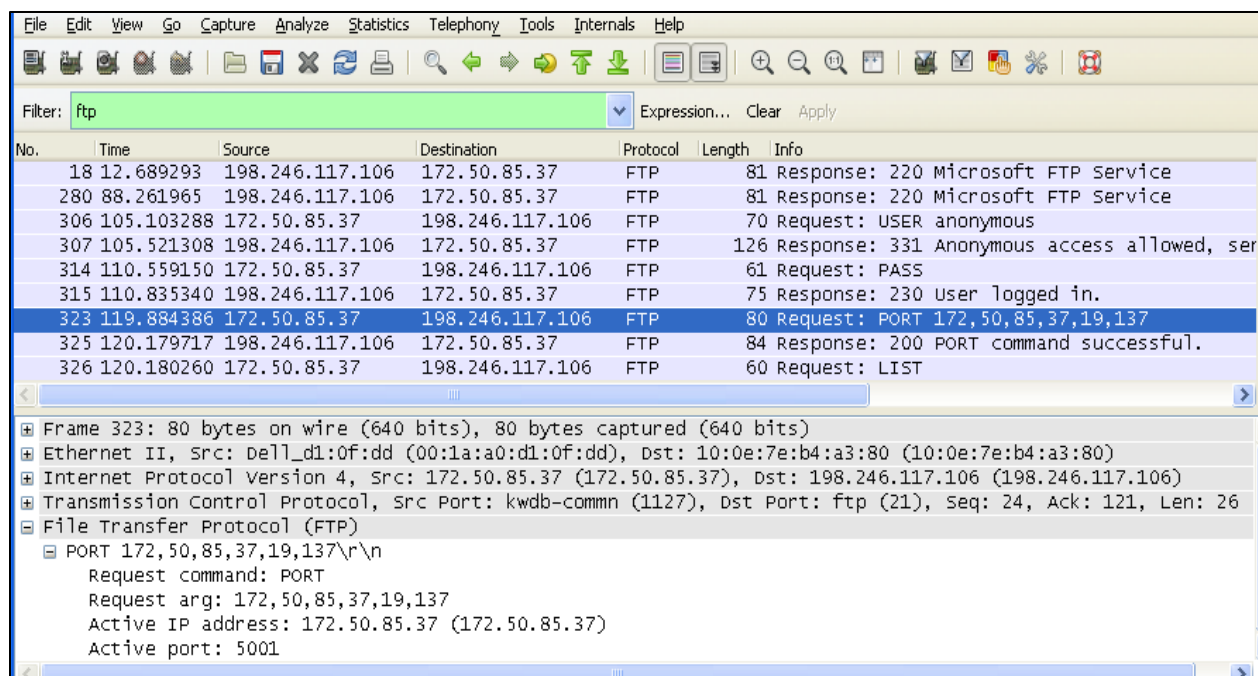


Figure 5

Step 3: **Stop the Wireshark capture.**

Step 4: **View the Wireshark Main Window.**

- Wireshark captured many packets during the FTP session to ftp.cdc.gov. The IP address, 198.246.117.106, is the address for ftp.cdc.gov.
- You can see the various protocols working underneath FTP in Wirshark (ie TCP, IP etc).
OR go to **Statistics** → **Protocol Hierarchy** (see Figure No-6) to find the same.

Step 5: **FTP Flow graph in Wireshark**

- Select a particular FTP stream.
- Go to **Statistic** Tab (see Figure No-6) → **Flow Graph** (click it)
- You will get a window with options select the option given by default (see Figure No-7) and press **OK**.
- You will get the FTP flow graph (see Figure No-8), analyse the FTP request response using flow graph.

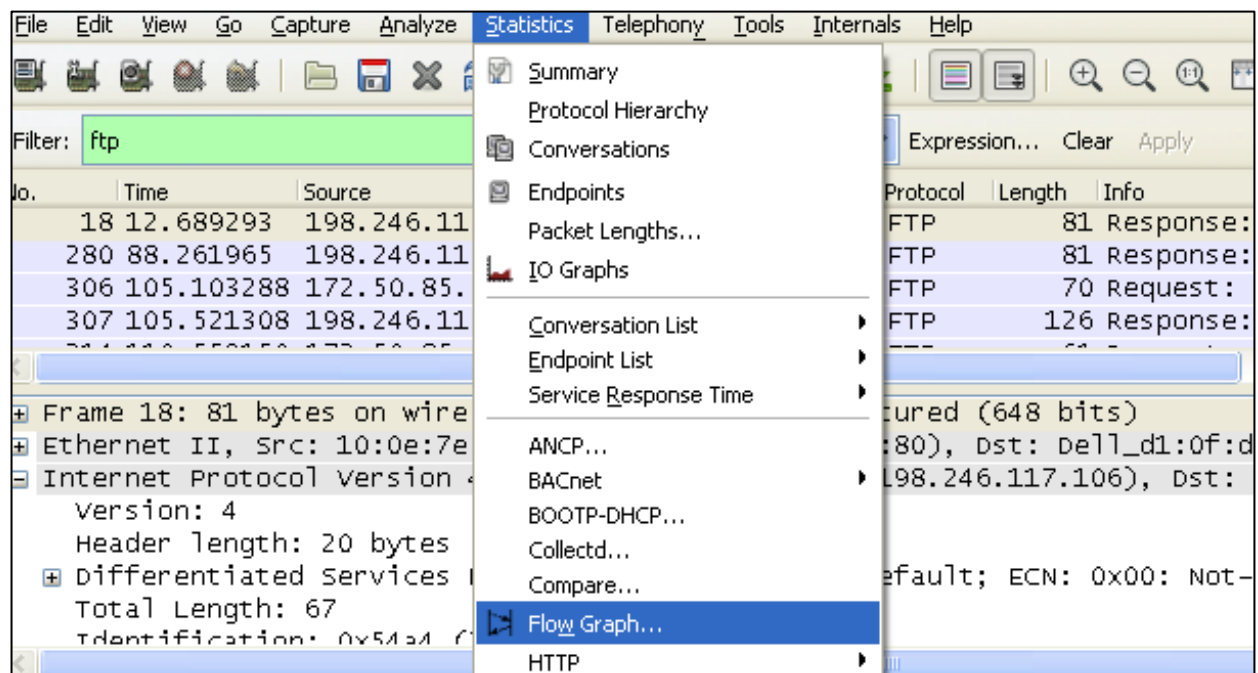


Figure 6

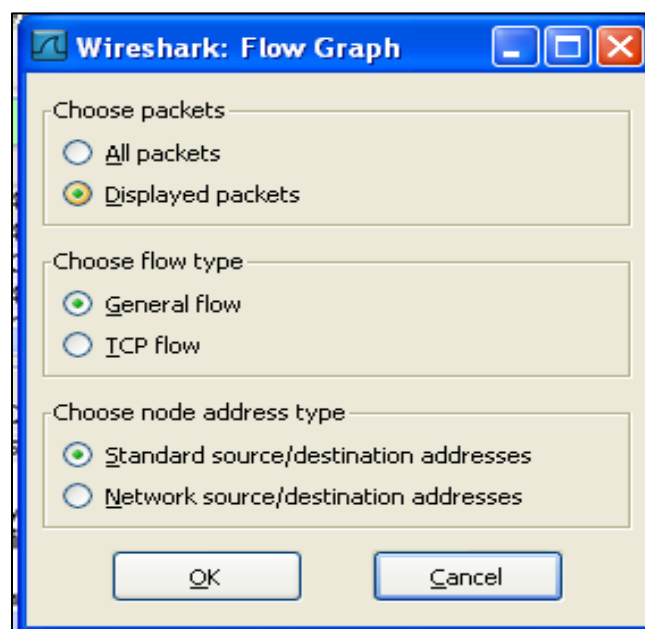


Figure 7

Time	198.246.117.106 172.50.85.37	Comment
12.689	(211) Response: 220 Micro (1119)	FTP: Response: 220 Microsoft FTP Service
88.262	(211) Response: 220 Micro (1127)	FTP: Response: 220 Microsoft FTP Service
105.103	(211) Request: USER anony (1127)	FTP: Request: USER anonymous
105.521	(211) Response: 331 Anony (1127)	FTP: Response: 331 Anonymous access allowed, send identity (e-mail name) as password
110.559	(211) Request: PASS (1127)	FTP: Request: PASS
110.835	(211) Response: 230 User (1127)	FTP: Response: 230 User logged in.
119.884	(211) Request: PORT 172,5 (1127)	FTP: Request: PORT 172,50,85,37,19,137
120.180	(211) Response: 200 PORT (1127)	FTP: Response: 200 PORT command successful.
120.180	(211) Request: LIST (1127)	FTP: Request: LIST
120.481	(211) Response: 150 Openi (1127)	FTP: Response: 150 Opening ASCII mode data connection.
141.200	(211) Response: 550 (1127)	FTP: Response: 550
159.376	(211) Request: PORT 172,5 (1127)	FTP: Request: PORT 172,50,85,37,19,138
159.711	(211) Response: 200 PORT (1127)	FTP: Response: 200 PORT command successful.
159.711	(211) Request: LIST (1127)	FTP: Request: LIST
160.049	(211) Response: 150 Openi (1127)	FTP: Response: 150 Opening ASCII mode data connection.
160.684	(211) Response: 550 (1127)	FTP: Response: 550

Figure 8

-End-