Hands-on Lab Description

CS-SYS-00006 – FTP (vsFTP) Setup on Linux

CONTENTS

Category:

```
CS-SYS: Computer System
```

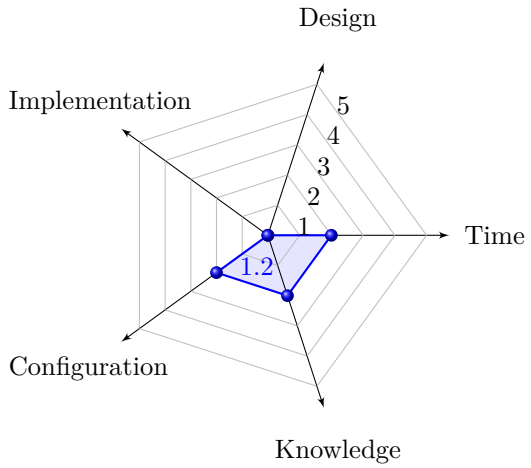
Objectives:

```
Learn File Transfer Protocol (FTP) basis
Learn how to set up vsftpd
```

Estimated Lab Duration:

```
Expert: 20 minutes
Novice: 100 minutes
```

Difficulty Diagram:



Difficulty Table.

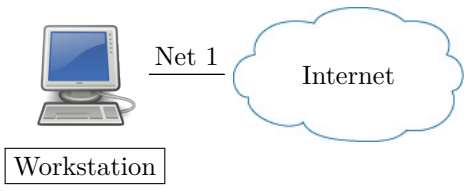
Measurements	Values (0-5)
Time	2
Design	0
Implementation	0
Configuration	2
Knowledge	2
Score (Average)	1.2

Required OS:

```
Linux: Ubuntu 18.04 LTS
```

Lab Running Environment:

```
ThoTh Lab: https://thothlab.org
```



```
1 Server: Linux (Ubuntu 18.04 LTS)
2 Network Setup: connected through a local network
```

Lab Preparations:

```
Initial setup: basic Ubuntu 18.04 LTS is required for this lab
Basic Linux knowledge and operations. Reference Lab: CS-SYS-00001.
```

File Transfer Protocol (FTP) is a TCP protocol for downloading files between computers. FTP works on a client/server model. The server component is called an FTP daemon. It continuously listens for FTP requests from remote clients. When a request is received, it manages the login and sets up the connection. For the duration of the session it executes any of commands sent by the FTP client.

Access to an FTP server can be managed in two ways:

```
1 Anonymous
2 Authenticated
```

In the Anonymous mode, remote clients can access the FTP server by using the default user account called "anonymous" or "ftp" and sending an email address as the password. In the Authenticated mode a user must have an account and a password. This latter choice is very insecure and should not be used except in special circumstances. If you are looking to transfer files securely see SFTP and refer to the labs on OpenSSH-Server. User access to the FTP server directories and files is dependent on the permissions defined for the account used at login. As a general rule, the FTP daemon will hide the root directory of the FTP server and change it to the FTP Home directory. This hides the rest of the file system from remote sessions.

Note: in ThoTh lab project running environment, you can set up an ftp service on any given VM to ease the data and file transfer between VMs.

1 Task 0 Preparation of FTP service

You can use *vsftpd*, which is an FTP daemon available in Ubuntu. It is easy to install, set up, and maintain. To check if the system has the *vsftpd* installed, issue the following command:

```
$    vsftpd -v
```

If not, you can install *vsftpd*. At a terminal prompt, you can enter the following command:

```
$    sudo apt install vsftpd
```

2 Task 1 Setup vsftpd

By default *vsftpd* is not configured to allow anonymous download. If you wish to enable anonymous download edit */etc/vsftpd.conf* by changing:

```
anonymous_enable=Yes
```

An example of vsftpd configuration file that enabled the anonymous access is given as follows:

```
# allow local users to log in.
local_enable=YES
#
# enable any form of FTP write command.
write_enable=YES
#
# allow the anonymous FTP user to upload files. This only
# has an effect if the above global write enable is activated. Also, you will
# obviously need to create a directory writable by the FTP user.
```

```
anon_upload_enable=YES
#
#
# For active mode, make sure PORT transfer connections originate from port 20 (ftp-data).
connect_from_port_20=YES
#
# no password is needed for anonymous mode
no_anon_password=YES
#
# anonymous user access folder
anon_root=/srv/ftp
#
```

During installation of vsftp, it creates a home directory of */srv/ftp*. This is the default FTP directory. When using anonymous mode, the user may experiencing 553 errors when uploading a file on to the ftp server. This is due to the write permission is not set for anonymous users to create a file on the server. In order allow any user to access and upload a file into the directory, you may want to add write privilege on this folder by issuing:

```
$    sudo chmod 755 /srv/ftp % Note that this is not a good practice for security
       reason that anyone can write on this folder.
```

If you wish to change this location, to */srv/files/ftp* for example, simply create a directory in another location and change the ftp user's home directory (this step is optional):

```
$    sudo mkdir /srv/files/ftp
$    sudo usermod -d /srv/files/ftp ftp
```

After making the change restart *vsftpd* by using any of the following two approaches:

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

Finally, copy any files and directories you would like to make available through anonymous FTP to */srv/-files/ftp*, or */srv/ftp* if you wish to use the default.

To test the ftp running status locally, you can first to ensure the daemon is running by using netstat

```
$    netstat -plan | grep :21
```

Or, you can check your vsftp running status:

```
$    service vsftpd status
```

You can also connect locally by making a connection to the localhost or 127.0.0.1

```
$    ftp localhost
$    ftp ip_address % test from a different machine to your server's IP address
```

3 Task 2 Setup Active or Passive Modes of FTP Service

Active and passive modes in FTP are the two connection modes it can communicate with. Note that only Linux currently support passive mode, and Windows OS does not support passive mode. Their basic service messaging sequences are presented in Figure CS-SYS-00006.1. FTP is somewhat unique in that it uses two channels between client and server, the command channel and the data channel, which are usually on separate

TCP connections. Typically the command channel is on port 21 and the data on port 20. In the figure, these two ports are considered well-known ports and they are reserved for FTP server. Other ports are randomly selected during the FTP establishment procedure. During the initial ftp-session establishment, the command channel handles the delivery of commands and responses typically, the data channel handles the actual transfer of files.

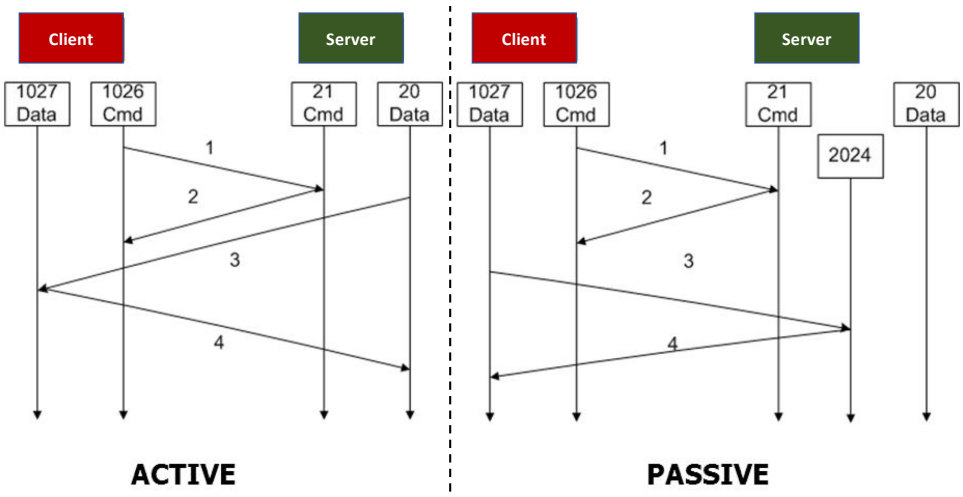


Figure CS-SYS-00006.1
Active and Passive FTP services.

The difference between active and passive FTP connections lies in whether the server or the client initiate the data connection. In active mode, the server initiates the data connection with the client after the client has established a connection on the command channel, i.e., the server initiates the data channel connection to port 1027 on the client. In passive mode it is the other way around, the client initiates the data connection with the server, i.e., the client initiates the data channel connect to the port 2024 on the server. In these two scenarios, both port 1027 and port 2024 are negotiated through the established command channel.

Depends on where the firewall NAT service is enabled, active or passive mode ftp may not successfully establish a data channel. For example, active FTP may fail in cases where the client is behind a firewall and protected from many to one NAT (masquerading). This is because the firewall will not know which of the many servers behind it should receive the data connection initiated from the server. On the contrary, when the NAT service is set on the server side, the passive mode may fail since the firewall may reject a data channel connection initiated by the client to a randomly chosen port 2024 on the service side.

By default, *vsftpd* is configured as active mode. To enable the passive model, edit */etc/vsftpd.conf* and add the following lines:

```
pasv_enable=Yes
pasv_max_port=10100
pasv_min_port=10090
```

The range specified in [*pasv_min_port, pasv_max_port*] provides the server-side port ranges that the client can connect to for establishing data-channel connections.

After making the change restart *vsftpd*:

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

To access ftp server using passive mode, you can issue:

```
$    ftp -p ftp_IP
```

Alternatively, after logging in the ftp server, you can issue one of the following commands to access via the passive mode:

```
ftp> quote PASV
ftp> passive
```

4 Task 3 Securing FTP

By default v*sftpd* is configured to authenticate system users and allow them to download files. If you want users to be able to upload files, edit */etc/vsftpd.conf*:

```
write_enable=YES   % allow upload files for authenticated users
anon_upload_enable=YES % allow upload files for anonymous users, which is risky and
    used carefully.
```

Then restart *vsftpd*:

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

The configuration file consists of many configuration parameters. The information about each parameter is available in the configuration file. Alternatively, you can refer to the man page, by running *man vsftpd.conf* for details of each parameter.

There are options in */etc/vsftpd.conf* to help make *vsftpd* more secure. For example:

```
chroot_local_user=YES
% uncommenting this, users can be limited to their home directories
chroot_list_enable=YES
chroot_list_file=/etc/vsftpd.chroot_list % uncommenting this, limit a specific list
    of users to just their home directories, create a /etc/vsftpd.chroot_list
    containing a list of users one per line
```

Then restart vsftpd:

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

Also, the */etc/ftpusers* file is a list of users that are disallowed FTP access. The default list includes *root*, *daemon*, *nobody*, etc. To disable FTP access for additional users simply add them to the list.

FTP can also be encrypted using FTPS. Different from SFTP, FTPS is FTP over Secure Socket Layer (SSL). SFTP is a FTP like session over an encrypted SSH connection. A major difference is that users of SFTP need to have a shell account on the system, instead of a *nologin shell*. Providing all users with a shell may not be ideal for some environments, such as a shared web host. However, it is possible to restrict such accounts to only SFTP and disable shell interaction.

To configure FTPS, edit */etc/vsftpd.conf* and at the bottom add:

```
ssl_enable=Yes
```

Also, notice the certificate and key related options:

```
rsa_cert_file=/etc/ssl/certs/ssl-cert-snakeoil.pem
rsa_private_key_file=/etc/ssl/private/ssl-cert-snakeoil.key
```

By default, these options are set to the certificate and key provided by the *ssl-cert* package. In a production

environment these should be replaced with a certificate and key generated for the specific host. Now restart *vsftpd*, and non-anonymous users will be forced to use FTPS:

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

To allow users with a shell of */usr/sbin/nologin* access to FTP, but have no shell access, edit */etc/shells* adding the *nologin shell*:

```
# /etc/shells: valid login shells
/bin/csh
/bin/sh
/usr/bin/es
/usr/bin/ksh
/bin/ksh
/usr/bin/rc
/usr/bin/tcsh
/bin/tcsh
/usr/bin/esh
/bin/dash
/bin/bash
/bin/rbash
/usr/bin/screen
/usr/sbin/nologin
```

This is necessary because, by default *vsftpd* uses *PAM* for authentication, and the */etc/pam.d/vsftpd* configuration file contains:

```
$    auth  required pam_shells.so
```

The shells PAM module restricts access to shells listed in the */etc/shells* file. Most popular FTP clients can be configured to connect using FTPS. The *lftp* command line FTP client has the ability to use FTPS as well.

5 Related Information and Resource

```
See the vsftpd website for more information.
For detailed /etc/vsftpd.conf options see the vsftpd.conf man page.
```