**NETW240 Week 5 Lab: Secure FTP and Secure Shell (SSH)**

**Lab Scenario**

**Assignment**

The purpose of this lab is to

* configure vsFTP server in Linux.
* learn about secure shell (SSH), and secure copy (SCP).

**Virtual Machine Login Information for PLABFED01 and PLABFED02**

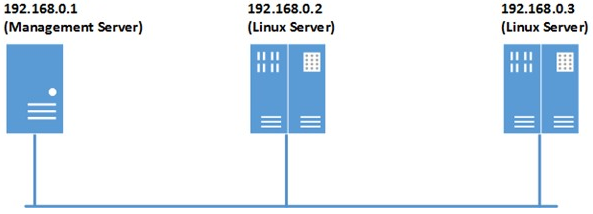
Username: **Student**   
Password: **Password**

Username: **root**   
Password: **Password**

**Lab Diagram**

During your session you will have access to the following lab configuration.

**WINCONSOLE PLABFED01 PLABFED02**



The Linux servers also connect to a private network. The **IP address of PLABFED01 is 192.168.240.11 and.** The **IP address of PLABFED01 is 192.168.240.12.**

**Connecting to your lab**

In this module you will be working on the following equipment to carry out the steps defined in each exercise.

* **WINCONSOLE (Management Server)**
* **PLABFED01 (Linux Server)**
* **PLABFED02 (Linux Server)**

Each exercise will detail which console you are required to work on to carry out the steps.

To start simply click on the named Server from the device list (located on the left hand side of the screen) and click the **Power on** from the in tools bar. In some cases the devices may power on automatically.

During the boot up process an activity indicator will be displayed in the name tab:

* **Black - Powered Off**
* **Orange - Working on your request**
* **Green - Ready to access**

If the remote console is not displayed automatically in the main window (or popup) click the **Connect** icon located in the tools bar to start your session.

If the remote console does not appear please try the following option:

* **Switch between the HTML 5 and Java client versions in the tools bar.**

In the event this does not resolve your connectivity problems please visit our Help / Support pages for additional resolution options.

###### **PART 1 vsFTP server configuration:**

###### **TASK A – Read General Background Information on vsFTP server:**

This is the background information to help you do your lab. Just read to have a general understanding how to configure a VsFTP server.

* FTP, File Transfer Protocol, is a client/server protocol for sharing files between machines over a TCP/IP network. This lab covers the popular **vsFTP** program which comes standard with Fedora 20.
* There are two kinds of FTP logins that vsFTP provides:
* anonymous FTP - login with the username '**anonymous**'
* real FTP - login with a real username and password.

**Familiar with vsFTP files**

Setting up the server will be done in two steps: Setting up the configuration files for vsFTP, and then starting the vsFTP services.

1. ***Setting up the Configuration Files***

There are three main configuration files you will need to edit to set up an VSFTPD server:

* /etc/vsftpd/vsftpd.conf
* /etc/hosts.allow
* /etc/hosts.deny

1. **/etc/vsftpd/vsftpd.conf file**: This file contains a list of directives which control the behaviors of each users. Below are directives summary tables

**Real-users vsftp control settings**

|  |  |
| --- | --- |
| local\_enable=YES | Allow any real-user ftp access |
| userlist\_enable=YES | Enable the denied access list |
| write\_enable=YES | Enable any form of FTP write command |
| local\_umask=022 | Umask for local users |

**Anonymous users vsftp control settings**

|  |  |
| --- | --- |
| anonymous\_enable=YES | Allow anonymous FTP |
| anon\_upload\_enable=YES | Allow the anonymous FTP user to upload files |
| anon\_umask=022 | Control the permissions of upload files |
| anon\_other\_write\_enable=YES | Allow anonymous users to rename or delete any files |
| anon\_mkdir\_write\_enable=YES | Allow anonymous users to create their own directories |

**ASCII enable settings**

|  |  |
| --- | --- |
| ascii\_download\_enable=YES | Allow download in ASCII mode |
| ascii\_upload\_enable=YES | Allow upload in ASCII mode |

**Logging vsftpd activities**

|  |  |
| --- | --- |
| xferlog\_enable=YES | Activate logging of uploads/downloads |
| xferlog\_file=/var/log/vsftpd.log | You may override where the log file goes if you like |

1. **/etc/hosts.allow and /etc/hosts.deny**

These two files specify which **computers** on the network can use services on your machine. Each line of the file contains a single entry listing a service and a set of machines.

When the server gets a request from a machine, it first checks hosts.allow then checks hosts.deny. It denies access to the matched entries in hosts.deny except the matched entries listed in hosts.allow.

* **/etc/hosts.deny**

To denying everybody:

|  |
| --- |
| vsftpd:ALL |

* **/etc/hosts.allow**

To allow access from localhost:

|  |
| --- |
| vsftpd: 127.0.0.1 |

To allow access from everybody on the 192.168.1.0/24 subnet:

|  |
| --- |
| vsftpd: 192.168.1. |

To allow access from two specific addresses:

|  |
| --- |
| vsftpd: 192.168.1.100 192.168.5.53 |

To allow access to everybody:

|  |
| --- |
| vsftpd: ALL |

1. ***Starting the vsFTP services***

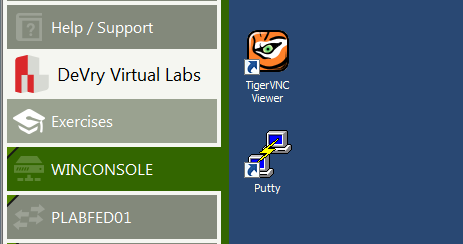
|  |
| --- |
| #systemctl start vsftpd |

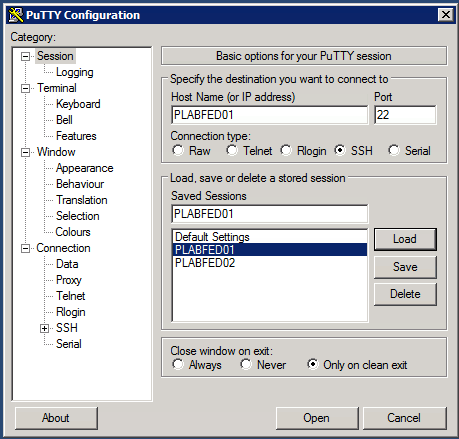
Note: How to stop, start, and restart vsftpd:

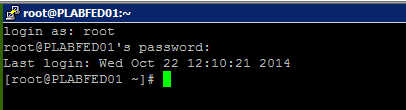
|  |  |
| --- | --- |
| Command | Description |
| #systemctl stop vsftpd | stop vsftpd |
| #systemctl start vsftpd | start vsftpd |
| #systemctl restart vsftpd | restart |
| #systemctl status vsftpd | Show status |

**TASK B – Configure Basic Settings For vsftpd Step-by-step**

We start setting up a basic vsFTP server. As always when doing any type of system administration, we need to perform them using root account. Use Putty to **logon plabfed01 as root**.







**Step 1. To** stop all firewalls or security features type the following commands:

|  |
| --- |
| #systemctl stop firewalld  #systemctl stop iptables  #setsebool -**P** ftp\_home\_dir=1 |

**Step 2.** Make sure the vsftpd packages are installed. You should be able to discover this with the following command:

|  |
| --- |
| #rpm -q vsftpd |

If it tells you that vsftpd is not installed, to install it type:

|  |
| --- |
| #yum –y install vsftpd |

**Step 3.** The file /etc/vsftpd/vsftpd.conf controls the configuration of vsftp. </SPAN> Use **nano or vi** editor to remove the # symbol before each directive if they are not done so and set it to YES or NO. We only go over some directives below so you understand how to customize it and leave the others at the default setup.

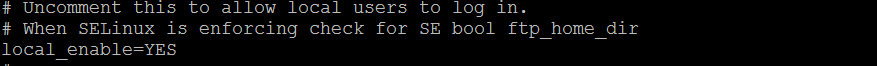
|  |
| --- |
| #nano /etc/vsftpd/vsftpd.conf |

* To disable access to your vsftpd server for **anonymous users**, change the directive as seen below:



Note: To enable the above feature, set anonymous\_enable=YES.

* To allow any user with **real-user** accounts on your system to access the vsftp server, change the directive as seen below:



* To allow real-user write access, so that they can upload material and modify content:

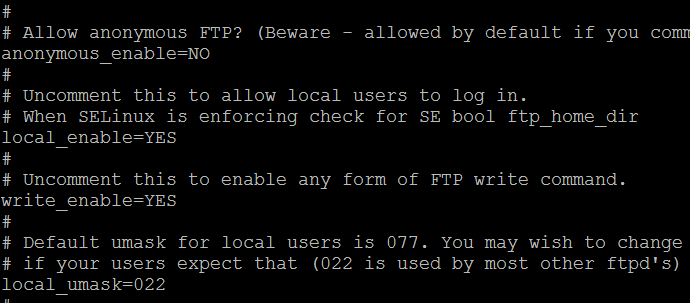


* All user accounts listed in the file /etc/vsftpd/user\_list are denied access. The following line in the /etc/vsftpd/vsftpd.conf activates that list. </SPAN>



Check the /etc/vsftpd/user\_list file to see which users are denied access to the vsftpd server.

Below is a partial list of the /etc/vsftpd/vsftpd.conf file.



Note: Lines preceded by the # symbol are comments.

**<SPAN class=bnote>Step 4.** After modifying the /etc/vsftpd/vsftpd.conf file, s</SPAN>tart the vsftpd service.

|  |
| --- |
| #systemctl start vsftpd |

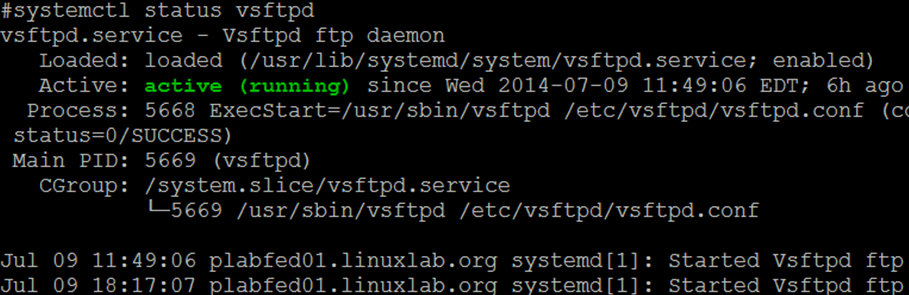
**Note**: If the file /etc/vsftpd/vsftpd.conf is modified, to make it effective you

need to restart the vsftpd. </SPAN>

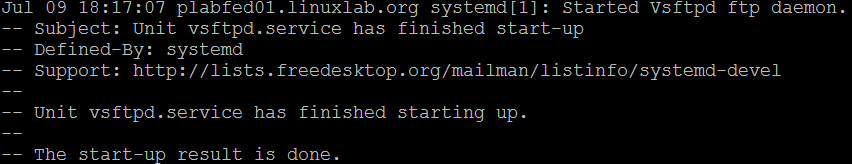
###### **Step 5**. To check **vsftpd** statustype:

|  |
| --- |
| #systemctl status vsftpd |

An active vsftpd status looks like this.



**Note:** If vsftpd fails starting, check for errors using command *journalctl –xn,* fix any typo in the /etc/vsftpd/vsftpd.conf file and restart the service with command #systemctl restart vsftpd until vsftpd starts successfully. Below is a sample of *journalctl –xn* command output without any error.



**Step 6.** </SPAN>To enable the vsftp server during boot, type the following command:

|  |
| --- |
| #systemctl enable vsftpd.service |

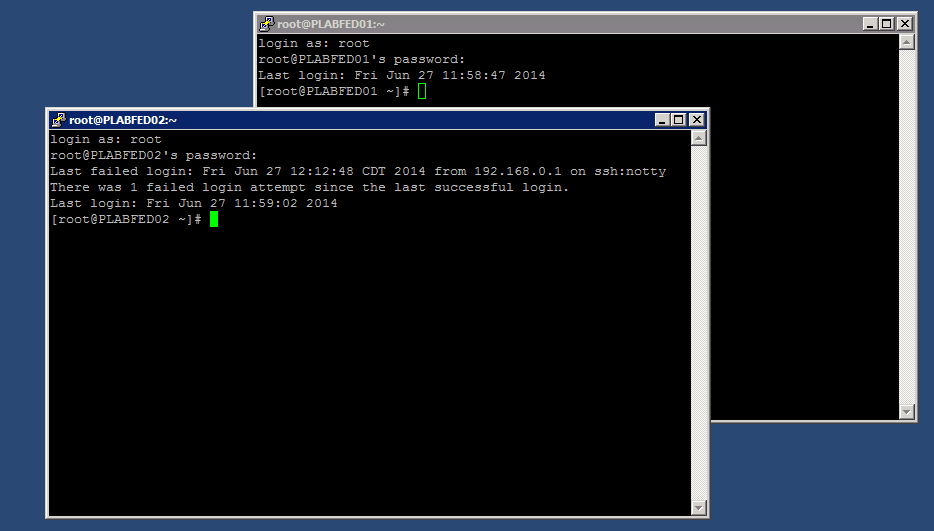
**TASK C –Validation and Testing**

###### Now we need to have two Putty sessions, one logon **plabfed01** and the other on **plabfed02**

###### Use Putty to logon plabfed01 as root.

* Use Putty to logon plabfed02 as root.

Arrange the terminal so we can switch back and forth easily. See below screenshot:



###### **Step 1**. Create a user account on **plabfed01** (vsftp server) using these commands with the password Password:

|  |
| --- |
| #useradd usera  #passwd usera |

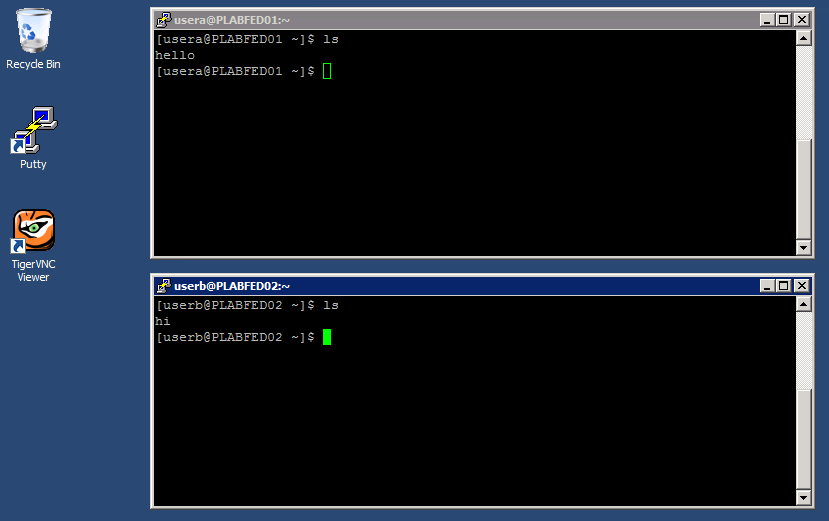
###### **Step 2**. Create a user account on **plabfed02** (ftp client) using these commands with the password Password:

|  |
| --- |
| #useradd userb  #passwd userb |

**Step 3.** Create a test file on plabfed01:Logon to usera on plabfed01 by using su – usera and create a file called hello (use *nano hello)* with the message “ This message is from userA” and save it.

**Step 4.** Create a test file on plabfed02:Logon to usera on plabfed02 by using su – userb and create a file called hello (use *nano hi)* with the message “ This message is from userB” and save it.

See the screenshot below:

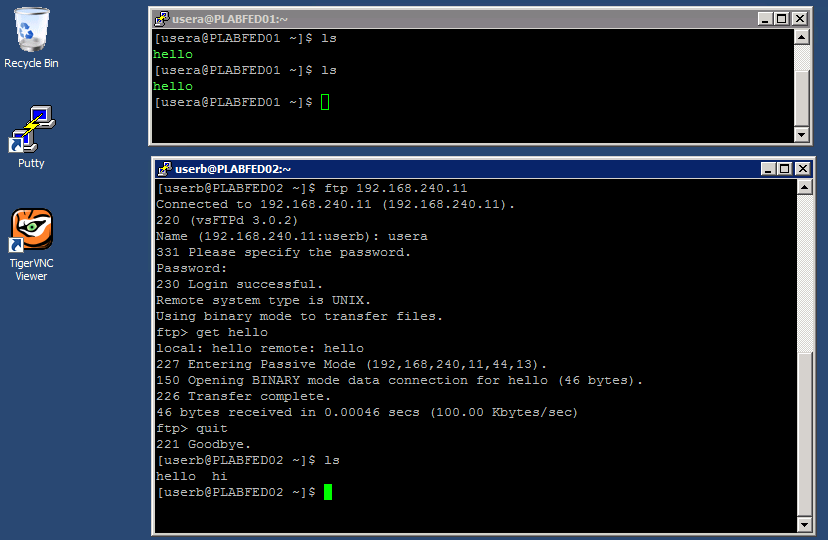


###### **Step 5**. We are going to use vsftp to upload the file “hi” from the userb account on **plabfed02** (ftp client) to usera account **plabfed01** (vsftp server). The result screenshot should look like this:

###### 

###### 

###### **Step 6**. We are going to use vsftp to download the file “hello” from the usera account on **plabfed01** (vsftp server) to userb account **plabfed02** (ftp client). The result screenshot should look like this:



###### **Step 7**. We are going to test if anonymous ftp logon worked. It should fail since we disable anonymous logon. We do this from the userb account on **plabfed02**. The result screenshot should look like this:

###### 

**PART 2 secure shell (SSH), and secure copy (SCP)**

**Scenario:**

* user Student has account on both plabfed01 and plabfed02 systems
* ssh software was installed on both systems and the sshd were started
* we want to logon plabfed01 as Student
* use ssh to logon plabfed02 as Student, create a file there
* use scp to copy back to plabfed01

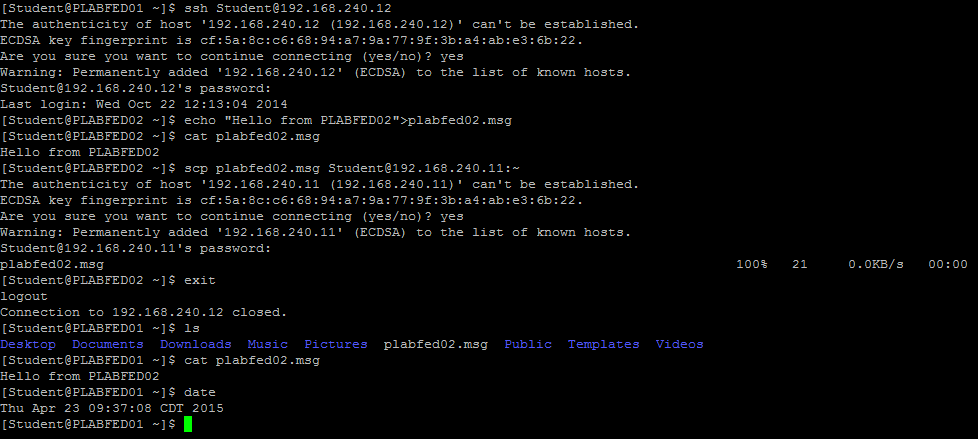
**Action:**

Use Putty to logon **plabfed01** using **Student** user. Clear the screen and complete the steps below.

1. **ssh to plabfed02** using **ssh Student@192.168.240.12**

or **ssh –l Student 192.168.240.12** (where *–l* is lowercase *L*)

1. Enter ***yes*** when you see, “Are you sure you want to continue connecting (yes/no)?”
2. Enter the Student password **Password** when prompted
3. Use echo to create the message ***Hello from plabfed02****,*redirecting the output to the file **plabfed02.msg**
4. Use the ***cat*** command to view the plabfed02.msg file contents
5. Copy the plabfed02.msg file to the Student home directory on plabfed01 by entering the command ***scp plabfed02.msg Student@192.168.240.11:~***
6. Exit the **ssh** remote connection and return to plabfed01
7. Use ***ls*** to display the files in /home/Student
8. Use the ***cat*** command to display the contents of the **plabfed02.msg** file
9. Use the ***date*** command to display the current date



**Result**   
You have gained experience with ssh and secure copy.

**PART 3 – Lab report preparation**

**Step 1**. Be sure to be in root account on **plabfed01**. Clear the screen and use the command

#tail -4 /etc/passwd file to display its last 4 lines and capture the Linux desktop and save this image to your Lab Report document in the space allocated for the #tail -4 /etc/passwd command.

**Step 2**. Be sure to be in **userb** account on **plabfed02**. Clear the screen, type the command #cat hi to display its contents and capture the Linux desktop and save this image to your Lab Report document in the space allocated for the hifile.

**Step 3**. Be sure to be in root account on **plabfed01**. Clear the screen and type the command #systemctl status vsftpd to display the vsftpd daemon status and capture the Linux desktop and save this image to your Lab Report document in the space allocated for the #systemctl status vsftpd command.

**Step 4**. Be sure to be in **userb** account on **plabfed02**. Clear the screen then ftp to 192.168.240.11 and logon as **usera**. Capture the Linux desktop and save this image to your Lab Report document in the space allocated for ftp and logon as usera.

**Step 5**. Be sure to be in **userb** account on **plabfed02**. Clear the screen then ftp to 192.168.240.11 and logon as **anonymous**. Capture the Linux desktop and save this image to your Lab Report document in the space allocated for ftp and logon as **anonymous**.

**Step 6**. Configure vsFTP allowing anonymous login and test it.

* Be sure to be in **root** account on **plabfed01**. Configure the vsFTP to allow **anonymous** login.
* Be sure to be in **userb** account on **plabfed02**. Clear the screen then ftp to 192.168.240.11 and logon as **anonymous**. Capture the Linux desktop and save this image to your Lab Report document in the space allocated for ftp and logon as **anonymous**.

**Step 7**. Configure vsFTP to **only disallow usera** login and test it.

* Be sure to be in **root** account on **plabfed01**. Configure the vsFTP to **only disallow usera** login.
* Be sure to be in **userb** account on **plabfed02**. Clear the screen then ftp to 192.168.240.11 and logon as **usera**. Capture the Linux desktop and save this image to your Lab Report document in the space allocated for ftp and logon as **usera**.

**Step 8.** Repeat Part 2 secure shell (SSH) and capture the lab desktop from plabfed01 showing the completion of all steps specified in part 2 above. Paste this image into the text box provided in your Lab Report document.

This concludes Lab 5.