Lab Instructions: Automating FTP Operations using pexpect

This document provides step-by-step instructions to automate FTP operations using the `pexpect` Python module.

Setting Up an FTP Server on Ubuntu- On your Remote machine

1. Update Your System

1. Before setting up the FTP server, let's update the package list on your Ubuntu system to ensure you have the latest updates.
2. Run the following commands:
   sudo apt update
   sudo apt upgrade -y

2. Install vsftpd

1. `vsftpd` (Very Secure FTP Daemon) is a popular and lightweight FTP server for Linux. Install it using the following command:
   sudo apt install vsftpd –y

vsftpd (Very Secure FTP Daemon) is used because it offers a high level of security, performance, and stability for managing FTP connections. It's designed to minimize vulnerabilities, supporting features like SSL/TLS encryption, chroot jails for user isolation, and access control, making it ideal for securely transferring files over FTP.

SSL (Secure Sockets Layer) and TLS (Transport Layer Security) - These protocols encrypt data transmitted between a client and a server, ensuring confidentiality, integrity, and authentication, preventing eavesdropping, data tampering, and impersonation.

3. Backup the Configuration File

1. Before making changes, back up the default `vsftpd` configuration file:
   sudo cp /etc/vsftpd.conf /etc/vsftpd.conf.bak

4. Configure vsftpd

1. Open the `vsftpd` configuration file in a text editor:
   sudo nano /etc/vsftpd.conf
2. Modify the following settings:
   - Allow local users to log in:
     Ensure the line below is uncommented:
     local_enable=YES

- Enable file upload and write permissions:
  Uncomment or add the following line:
  write_enable=YES
- Restrict users to their home directories:
  Uncomment or add the following line:
  chroot_local_user=YES

3. Save and exit the file **(Press `CTRL+O`, then `Enter`, and `CTRL+X`).**

5. Restart the FTP Service

1. After making changes, restart the `vsftpd` service to apply the configuration:
   sudo systemctl restart vsftpd
2. Enable the service to start on boot:
   sudo systemctl enable vsftpd

6. Create an FTP User

1. Add a new user account for FTP access:
   sudo adduser ftpuser
2. Follow the prompts to set a password and user details.
3. Assign the user a home directory:
   sudo mkdir -p /home/ftpuser/ftp/files
   sudo chown -R ftpuser:ftpuser /home/ftpuser/ftp/files
4. The `/ftp/files` directory is where the FTP user will upload/download files.
5. Restrict the user to their home directory (if not already enabled in the config):
   sudo usermod -d /home/ftpuser/ftp ftpuser

7. Adjust Firewall Rules

1. If you have a firewall enabled, allow FTP traffic using the following commands:
   sudo ufw allow 20/tcp
   sudo ufw allow 21/tcp
   sudo ufw reload

8. Test the FTP Server

1. Connect to the FTP server using the command-line of your **Local Machine.**
2. From your local machine, run:
3. ftp <your-remote-server-ip>
4. Enter the username (`ftpuser`) and password created in the previous step.
5. Incase if you are not able to connect to your ftp server using the credentials, feel free to change the password of the ftpuser, to do so, run:

   sudo passwd ftpuser

6. You will be prompted to enter the sudo user password first, then it will ask you to enter and confirm the new password

9. Creating a sample file on our FTP server.

Let's create a sample file in our FTP server (which is configured on our remote machine), so that we can use pexpect module to automate FTP operation (downloading the file from FTP server which we created in remote machine). Later on, we can verify that our FTP automation script in our local machine has successfully downloaded the file from FTP server.

1. In your remote machine, login as **ftpuser,** to do so, please enter the following command:

   su ftpuser
2. Enter the password that you created for ftpuser and press Enter.
3. You will be logged in as ftpuser, to verify that enter the following command:

   whoami
4. Check the present working directory, navigate to the following directory **/home/ftpuser/ftp/files** using **cd** command:

   cd /home/ftpuser/ftp/files
5. Now, in this directory, lets create a text file called remotefile.txt, to do so, we will use nano editor:

   nano remotefile.txt

   Enter the following text, **This is a remote FTP File.**

6. Then press **Ctrl+O** to save the changes, then press **Enter**, then **Ctrl+X** to close the nano editor.

7. Now, lets verify if we have created the remotefile.txt successfully, to do so, please enter the following command:

   ls

   Here, you can locate **remotefile.txt** file which we created.

10. Creating a sample file on our FTP localfile.txt.

Let's create a sample file on our local machine so that we can use pexpect module to automate FTP operation (uploading the file from our local machine to FTP server hosted in our remote machine). Later on, we can verify that our FTP automation script in our local

machine has successfully uploaded the file from our local machine to FTP server hosted on our remote machine.

8. Now, in this directory, lets create a text file called remotefile.txt, to do so, we will use nano editor:

   nano localfile.txt

   Enter the following text, **This is a local file using for FTP upload operation.**

9. Then press **Ctrl+O** to save the changes, then press **Enter**, then **Ctrl+X** to close the nano editor.

10. Now, lets verify if we have created the localfile.txt successfully, to do so, please enter the following command:

    ls

    Here, you can locate **localfile.txt** file which we created.

Step 9: Python Script for FTP Automation (execute the following script in your Local Machine's virtual environment)

1. In your local machine, activate the python virtual environment using the following command:

Source my_python_env/bin/activate

2. Now, lets create a python script with name **'Automating_FTP_Operations_using_Pexpect.py'** using nano editor. To do so, please enter the following command:

   nano Automating_FTP_Operations_using_Pexpect.py

3. Now copy paste the following python script to the file and replace the remote_host, 'username' and 'password' accordingly.

```
import pexpect

def ftp_automation(host, username, password, commands):
    try:
        # Spawn the FTP session
        ftp_command = f"ftp {host}"
        print(f"Connecting to FTP server: {host}")
        child = pexpect.spawn(ftp_command, timeout=30)

        # Handle login prompts
        child.expect("Name .*:")  # Match the prompt for username
        child.sendline(username)
        child.expect("Password:")  # Match the prompt for password
        child.sendline(password)

        # Check if login was successful
        index = child.expect(["ftp>", "Login incorrect", pexpect.EOF, pexpect.TIMEOUT])
        if index == 1:
            print("Login failed. Please check your username or password.")
            return
        elif index in [2, 3]:
            print("Error: Connection failed.")
            return

        print("Login successful. Executing commands...")

        # Execute each command
        for command in commands:
            print(f"Executing: {command}")
            child.sendline(command)
            child.expect("ftp>")
            output = child.before.decode()
            print(f"Output:\n{output}")

        # Close the FTP session
        child.sendline("bye")
        child.expect(pexpect.EOF)
        print("FTP session closed.")

    except pexpect.exceptions.TIMEOUT:
        print("Error: Operation timed out.")
    except Exception as e:
        print(f"An error occurred: {e}")

if __name__ == "__main__":
    # Replace with your FTP server details and commands
    ftp_host = "ftp_host_ip_address"
    ftp_username = "username"
    ftp_password = "Password"

    # List of commands to execute
    ftp_commands = [
        "ls",  # List files in the current directory
        "cd files",  # Change to the 'files' directory
        "put localfile.txt",  # Upload a file from the current working directory of your local machine
```

```
    "get remotefile.txt",  # Download a file from your FTP server
    "pwd",  # Print working directory
]

# Automate the FTP operations
ftp_automation(ftp_host, ftp_username, ftp_password, ftp_commands)
```

Then press **Ctrl+O** to save the changes, then press **Enter**, then **Ctrl+X** to close the nano editor.

Executing the script:

1. Ensure that your python virtual environment is activated.
2. To execute the python script please enter the following command:

   python Automating_FTP_Operations_using_Pexpect.py

3. Once the script is executed successfully, you will be able to see the ouput of the commands we executed in **Automating_FTP_Operations_using_Pexpect.py** script.
4. To verify the file transfers, type **'ls'** and press **Enter,** you will be able to see remotefile.txt in your present working directory of the localmachine, which means you have successfully downloaded the file from FTP server to your local machine.
5. And to verify that upload operation is successful, in your remote machine login as ftpuser and locate localfile.txt in this directory **/home/ftpuser/ftp/files (in your remote machine).**

Notes

- Ensure that the `pexpect` library is installed in your Python environment. Install it using:

  ```bash
  pip install pexpect
  ```

- Replace placeholders in the script (e.g., `ftp_host`, `ftp_username`, `ftp_password`) with your FTP server details.