Lab Instructions: Automate Remote Package Installation with pexpect on Ubuntu

1. Ensure that you are in your python virtual environment, it is not active, use the

following command to activate it:

source my_python_env/bin/activate

Prerequisites

1. Ensure Python is Installed **(ignore if already installed)**

- Verify the installation:
  python3 --version

2. Install Required Dependencies **(ignore if already installed)**

- Install `pexpect` for SSH automation:
  pip install pexpect

- Install `nest_asyncio` to handle nested async loops:
  pip install nest_asyncio

3. Create and Activate a Virtual Environment **(ignore if already installed)**

- Create a virtual environment:
  python3 -m venv venv

- Activate the virtual environment:
  source venv/bin/activate # For Linux/macOS

Step 2: Execute the Python Script

1. Now, lets create a python script with name **'Automate_package_download_on_remote_host.py'** using nano editor. To do so,

please enter the following command:

nano Automate_Remote_Host_Login_using_Paramiko.py

2. Paste the following code in the **Automate_Remote_Host_Login_using_Paramiko.py** file.

```
import asyncio
import pexpect
import nest_asyncio

async def ssh_connect(host, username, password):
    """Connect to a remote host via SSH using pexpect and handle unknown host key prompt."""
    ssh_command = f"ssh {username}@{host}"
    child = pexpect.spawn(ssh_command, timeout=None)
    try:
        index = child.expect(["Are you sure you want to continue connecting", "password:", pexpect.EOF, pexpect.TIMEOUT],
timeout=10)
        if index == 0:
            child.sendline("yes")
            child.expect("password:")
            child.sendline(password)
        elif index == 1:
            child.sendline(password)
        child.expect("[$#>] ", timeout=None)
        return child
    except pexpect.exceptions.TIMEOUT:
        print("Connection timed out.")
        return None

async def install_package(child, package_name, sudo_password):
    """Install a package on the remote machine."""
    command = f"echo {sudo_password} | sudo -S apt-get install -y {package_name}"
    child.sendline(command)
    try:
        index = child.expect(["Do you want to continue? [Y/n]", "password for", "[$#>] "], timeout=None)
        if index == 0:
            child.sendline("Y")  # Confirm installation
            child.expect("[$#>] ", timeout=None)  # Wait for prompt again
        elif index == 1:
            child.sendline(sudo_password)  # Send sudo password
            child.expect("[$#>] ", timeout=None)  # Wait for prompt again
        print(f"Installed {package_name} on the remote machine.")
    except pexpect.exceptions.TIMEOUT:
        print(f"Timeout while installing {package_name}. Check network or try again.")

async def main():
    host = input("Enter the remote host IP or domain: ")
    username = input("Enter the SSH username: ")
    password = input("Enter the SSH password: ")
    sudo_password = input("Enter the sudo password: ")

    child = await ssh_connect(host, username, password)
    if child is None:
        return

    package_name = input("Enter the package name to install: ")
    await install_package(child, package_name, sudo_password)

    child.sendline("exit")
    child.close()

if __name__ == "__main__":
```

```
nest_asyncio.apply()  # Apply nest_asyncio patch
asyncio.run(main())
```

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

2. Execute the Script

- Run the script using Python 3:
  python3 automate_package_installation.py

3. Provide Input Parameters

- Enter the remote host's IP or domain.

- Provide the **SSH username and password** when prompted.

-Provide the sudo password for package installation.

- Enter the package name you want to install.(ex: git)

-

Step 3: Verify Operations

- After the package installation, check the output to ensure that the package was successfully installed.

- You can log into the remote machine and verify the installation using the following command:
  which package