Lab Instructions: Login into Remote Host using Key with pexpect

In this lab log in to a remote host using a generated SSH key with the `pexpect` Python module.

Using SSH keys for remote login enhances security by replacing passwords with cryptographic keys, making it resistant to brute-force attacks. It also enables passwordless, streamlined access, saving time and improving convenience. Additionally, SSH keys offer better control and auditability, allowing easier management of access and automation.

Step 1: SSH Key Generation

1. Generate an SSH key pair on your local machine using the following command:

```bash
ssh-keygen -t rsa -b 2048 -f ~/.ssh/my_ssh_key  (you can replace ‘my_ssh_key’ with your name)
```

2. When it asks for any paraphrase, press enter and then enter.

The above command generates a private key (`~/.ssh/my_ssh_key`) and a public key (`~/.ssh/my_ssh_key.pub`).

3. To verify, please enter the following command to verify if the keys are generated.

```bash
cd /home/rps/.ssh
```

4. You will be able to see 2 keys created, one is **my_ssh_key**(this is private key) and the other is **my_ssh_key.pub** (public key)
5. Now lets switch our current working directory to our home directory, to do so, please enter the following command.

```
cd ..
```

The **private key** is kept secure on your local machine and used to prove your identity, while the **public key** is shared with the remote server to allow it to verify your identity. This system ensures secure, encrypted communication without exposing your private key to potential threats.

Step 2: Add the Public Key to the Remote Host

6. Now, let's copy the public key to the remote host to enable key-based authentication. To do so, please enter the following command in the terminal of your local host VM.

```bash
ssh-copy-id -i ~/.ssh/my_ssh_key.pub user@remote_host
```

Replace 'my_ssh_key' with the name you have given for the key in the previous step, `user` with the username and `remote_host` with the IP address or domain of the remote host.

Step 3: Test SSH Key Authentication

7. Verify that you can log in to the remote host without a password by running the following command in the terminal of your local host VM:

```bash
ssh -i ~/.ssh/my_ssh_key user@remote_host
```

If successful, this confirms that the key-based authentication is correctly set up.

8. To verify you are successfully logged in, we will just check the IP address of the logged in machine, to do so, please enter the following command:

ifconfig

You can locate the IP address of your remote machine, it means you have successfully loggedd in using SSH key.

9. To close the remote host connection, please enter the following command:

    Exit

    You will see the output stating that **'Connection closed'**

10. Now, you are in your local host, to verify type the following command:

    ifconfig

    You will see the IP address of the local host.

Step 4: Python Script using pexpect

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 **‘Logging_into_Remote_Host_using_a_Key.py’** using nano editor. To do so, please enter the following command:

   nano **Logging_into_Remote_Host_using_a_Key.py**

3. Now copy paste the following python script to the file and replace the remote_host, ‘username’ and ‘private_key_path’ accordingly.

```
import pexpect

# Define variables
remote_host = "remote_host"
username = "user"
private_key_path = "~/.ssh/my_ssh_key"

# SSH command
ssh_command = f"ssh -i {private_key_path} {username}@{remote_host}"

# Spawn the SSH session
child = pexpect.spawn(ssh_command, timeout=30)

# Handle SSH prompts
try:
    child.expect("Are you sure you want to continue connecting (yes/no/[fingerprint])?")
    child.sendline("yes")
except pexpect.exceptions.TIMEOUT:
    pass

# Interact with the shell
child.interact()
```

Notes

- Replace `user` and `remote_host` with the appropriate username and IP/domain of your remote host.

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

- This script will spawn an SSH session and allow you to interact with the remote host.

Executing the script:

1. Ensure that your python virtual environment is activated.

2. To execute the python script please enter the following command:

   ```
   python Logging_into_Remote_Host_using_a_Key.py
   ```

3. Wait for a couple of seconds for the program to execute, once the program is successfully executed, you will see that you are logged in to remote host.
4. To verify whether you are logged into remote host, please enter the following command:

   ```
   ifconfig
   ```

   If you can locate the IP address of the remote machine, which means you are successfully logged in to remote host using the key.

5. Enter **‘exit’** to close the remote connection