Exercise 5.5: Parallel ssh command execution

Configure and test the **pssh** command on the local adapters on your system.

The pssh or parallel-ssh command will send commands to many machines controlled by a text file as to what machines are used. **pssh** works best with StrictHostKeyChecking=no or previously added fingerprints to the `~/.ssh/knownhosts` file. The **pssh** commands are most secure with ssh key copied into the targets authorized_keys file.

Note: Some distros use the names like `pssh`, others use `parallel-ssh` to avoid conflicts with other software use the appropriate package management command to verify installation and the names being used.

Solution 5.5

1. Install or verify **pssh** is installed

   - **Ubuntu:**

     ```
     $ sudo apt-get update
     $ sudo apt-get install pssh
     ```

     Verify the program names

     ```
     $ dpkg-query -L pssh | grep bin
     ```

   - **CentOS:**

     ```
     $ sudo yum install pssh
     ```

     Verify the program names

     ```
     $ sudo rpm -ql  pssh | grep bin
     ```

   - **OpenSUSE:**

     ```
     $ sudo zypper install  pssh
     ```

     Verify the program names

     ```
     $ sudo rpm -ql  pssh | grep bin
     ```

2. Setup ssh keys and fingerprints If not already done, create a key pair on the local machine

   ```
   $ ssh-keygen
   ```

   Copy the key to the remote and save the fingerprint

   ```
   $ ssh-copy-id localhost
   ```

3. Test the paswordless connection, if you are prompted for anything fix it now

   ```
   $ ssh localhost
   ```

   Repeat for all the local interfaces or some remotes

   ```
   $ ssh-copy-id 127.0.0.1
   $ ssh-copy-id 172.16.104.135
   ```

4. Create a ip-list file for pssh

   ```
   $ echo "127.0.0.1"  > ~/ip-list
   $ echo "172.16.104.135" >> ~/ip-list
   $ echo "localhost" >> ~/ip-list
   ```