SSH Client and Server

To login to any machine with SSH, you need to install the ssh server. The client is installed on Debian by default. Terminology used is “Server” for the remote machine to get access to, and “Host” for the local machine which needs access to a server. This guide was updated 8th August 2020.

Linux Server

Installation

- `apt install ssh`
- Do the same for Windows Subsystem for Linux on Windows 10

Settings

- Modify /etc/ssh/sshd_conf:

```
Port 22
LoginGraceTime 20
AuthorizedKeysFile .ssh/authorized_keys
PasswordAuthentication no
PermitEmptyPasswords no
PermitRootLogin without-password
```

- If you want to allow ssh root access from a regular user on the same host, or from another host (e.g. BackupPC), add the following lines to the end of sshd_config:

```
Match Address my.host.subnet.ip
  PermitRootLogin without-password
```

- If you want to be able to access the X Server remotely, add the following to /etc/ssh/sshd_conf:

```
X11Forwarding yes
X11DisplayOffset 10
PrintMotd no
PrintLastLog yes
TCPKeepAlive yes
```

- Restart the SSH server:

```
sudo /etc/init.d/ssh restart
```

Linux Host

- Run ssh-keygen with the following command to generate a key pair.

```
ssh-keygen -t ecdsa -b 521
```

- Leave the password empty so you don't need to enter it every time when establishing a connection. Accept the defaults, which puts the generated key pair into ~/.ssh
- Copy the public key file (extension .pub) to all servers you need access to, then add the content of the file to the ~/.ssh/authorized_keys file
- The private key file (no extension) remains on the host and is never shared with anyone
- Observe permissions, folder .ssh, the private key file, and the authorized_keys file must have read/write permissions of the user only, all others must not have any access

How much data does SSH typically use?

Windows Host

- Install PuTTY with the installer or as PuTTY Portable app. Make sure to keep the installation updated.
- Run PuTTYgen to import or generate a key. Always use the updated version, which gets updated along with PuTTY.
- If you have an existing key pair generated on a Linux Host, then do the following:
  - Select **Conversions → Import key** in the menu
  - Modify the *Key comment* to <machine name>-<date>, which allows you to be able to identify the key and date when the key was generated
  - **Save private key**
  - Use the existing public key which was generated on the Linux Host
- Generate a key pair:
  - Select **ECDSA** as *Type of key to generate* at the bottom, select **nistp521** as *Curve to use for generating this key*, then click **Generate**
  - Modify the *Key comment* to <machine name>-<date>, which allows you to be able to identify the key and date when the key was generated
  - **Save public key** and **Save private key**

Relais Hosts

I have machines in a location without fixed IP address, and where external access is only possible through a relais host which disconnects after 2+ minutes of inacitivity. I solve this by adding the following to the sshd_config of each machine which need to be accessed.

```
ClientAliveInterval 300
ClientAliveCountMax 2
```

Alternatively, it could also be handled on the client side with the following lines in ssh_config, or keep alive setting in PuTTY.

```
Host *
  ServerAliveInterval 300
  ServerAliveCountMax 2
```

In addition I have restricted SSH access to machines with known IP addresses

- Find from where (relais machine) you connect through SSH

  ```
  $ who
  ```

- Add the restriction to your `authorized_keys` file

  ```
  restrict,from="aaa.bbb.ccc.ddd,eee.fff/16"

  ==== Links ====
    *
  [[https://patrickmn.com/aside/how-to-keep-alive-ssh-sessions/#:~:text=On%20Linux%20(ssh)&text=These%20settings%20will%20make%20the,to%20have%20been%20discarded%20anyway.|How to Keep Alive SSH Sessions]]
    *
  [[https://superuser.com/questions/1272875/relay-two-ssh-connections-together|Relay two SSH connections together]]
    *
  [[https://www.thethingsnetwork.org/docs/gateways/kerlink/reverse-ssh/|Reverse SSH]]
    *
  [[http://man.openbsd.org/sshd_config#:~:text=The%20client%20alive%20mechanism%20is,disconnected%20after%20approximately%2045%20seconds.|sshd_config — OpenSSH daemon configuration file]]
  ===== X Client =====
  ==== Debian ====
    * Establish an ssh connection from your graphical desktop to the
  remote X client using the “-X” switch for X11 forwarding:<code>ssh -X
  <user>@<Xclient>
  sensible-browser
  ```

Windows

- Install VcXsrv as X Server implementation with all defaults.
- Open PuTTY and establish an SSH connection from Windows to the remote X host, making sure you enable X11 forwarding in Connection>SSH>X11. Check the X11 forwarding box, put in “localhost:0.0” for the display location and select the “MIT-Magic-Cookie” setting.
- Run the default browser from the remote host's command line with:

  ```
  sensible-browser
  ```

Links

- ssh-keygen: Generate a New SSH Key
- How To Set Up SSH Keys
- X11 forwarding to view GUI applications running on server hosts
- Headless Mode for Virtual Machines of VirtualBox
- LightDM
- Use Your SSH Config File to Create Aliases for Hosts