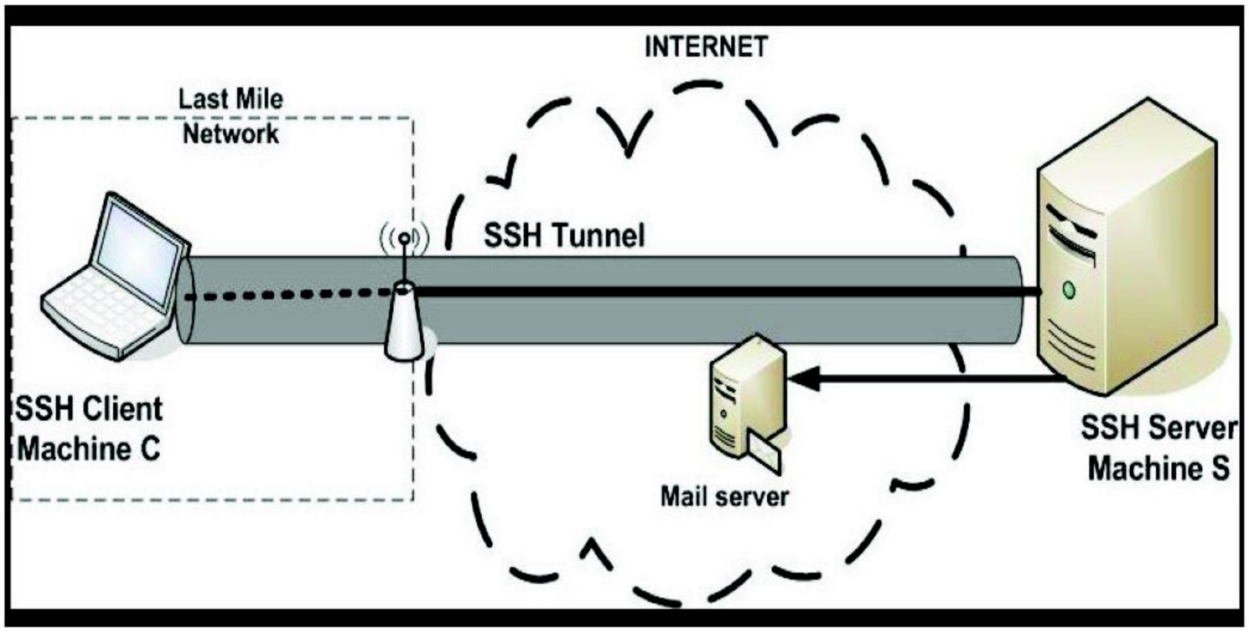
Secure **SH**ell **(SSH)**

What is SSH?

One essential tool to for a system administrator is SSH.
SSH, or Secure Shell, is a protocol used to securely log onto remote systems. It is the most common way to access remote Linux and Unix-like servers.



Getting started – install SSH :

First, update system and install necessary packages to our system.
To update the system and install the SSH server on the server machine, run the following command:

```
$ sudo apt-get update
$ sudo apt-get install openssh-server
```

To install SSH client on the client machine, we should run the following command:

```
$ sudo apt-get install openssh-client
```

<u>Getting connected with SSH :</u>

1. Connect with password : **$ ssh remote_username@remote_host**
2. Connect without password : **For this we have to follow some steps as stated below.**

Steps :

1. On our client machine, generate SSH keys with the following command:

```
$ cd  ~./ssh
$ ssh-keygen  -t  rsa
```

Simply press the Enter key at every prompt. This produces two files: id_rsa.pub (public key) and id_rsa (private key).

2. On our server, create the folder as below :

```
$ mkdir  -p  ~/.ssh/
```

3. Back to our client machine, copy the "**id_rsa.pub**" file to our server using the following command:

```
$ scp -P  "ourport"  ~/.ssh/id_rsa.pub  username@server_ip:~/.ssh
```

We can change "ourport" to the port number that your SSH server is using (the default is **22**) and the "serverip" to the server's IP address.

4. On our server machine, change the filename and setup permissions.

```
$cat   ~ /.ssh  /id_rsa.pub >>  ~ /.ssh/ authorized_keys
$ chmod  700   .ssh
$ chmod  600   .ssh/ authorized_keys
$ rm  .ssh/id_rsa.pub
```

5. To test if the key-based authentication method works, try connecting to our SSH server from the client machine:

```
$ ssh remote_username@remote_host
```

If we are able to connect without entering a password, then the key-based authentication method works.

Some *additional* configurations we can do :

Secure SSH configuration file : **"/etc/ssh/sshd_config"**

The "/etc/ssh/sshd_config" file is the system-wide configuration file for SSH which allows us to set different options to improve the security of an SSH . The default configuration in the config file is very insecure, so we need to edit it first and set proper options to improve the security.

1. Change SSH listening port :

Port **22**
to
port **2200.**

2. Version 1 of the protocol contains security vulnerabilities. Protocol 2 is the default entry on Ubuntu.

Protocol 2

3. Limit users access :
It is necessary to allow only specific users to log in to SSH. It can improve your security. By default, this option is not available in the SSH configuration file.
To allow **“user1**” and “**user2**,” add the following line:

Allowusers user1 user2

To deny “baduser1” and “baduser2,” add the following line:

Denyusers baduser1 baduser2

4. Disable root login: Change as mentioned below.

PermitRootLogin **without-password**
To
PermitRootLogin **no**

5. Hide last login : Change as mentioned below
You can hide who logged in last when a user logs in.

PrintLastLog **yes**
To
PrintLastLog **no**

6. Restrict the interface to log in: By default, **ssh** will listen on all network interfaces. If we want to allow an SSH connection to be accepted from **specific IP addresses**, we can change the line as mentioned below.

#ListenAddress ::
To
ListenAddress **ip_address**

7. Disable password authentication : Using password authentication is a big security risk if your user uses a weak password. It is recommended to use “ssh keys.” An “ssh key” can contain over **600** random characters and be difficult to break.

```
# PasswordAuthentication yes
                    To
PasswordAuthentication no
```

8. Set a login grace timeout : The “LoginGraceTime” specifies how long after a connection request the will wait before disconnecting. It is recommended to reduce it to 60 seconds.

```
LoginGraceTime 120
          To
LoginGraceTime 60
```

Now save and exit the **/etc/ssh/sshd_config** file and restart the SSH server.

```
$ sudo service ssh restart
```