

Session#4 Part 2



Managing a Remote Shell

SSH

It stands for Secure SHell

You can use it to connect to remote device.

To connect to a server you should know its address(ip) or name (DNS server will resolve the name to ip address) and have a user on it.

Usage : `ssh <user_name>@<host> -p <port_number>(default 22)`

Example:

Assume we have host : `127.0.0.1` and user : `user1` with password : `pass`

using **ssh** to connect to the server : `ssh user1@127.0.0.1`



SSH-KEYGEN

This command used to create a public and private key.

You can use those keys with **ssh** to connect the server instead of the password. It's more secure than password text.

Usage : `ssh-keygen -t <encryption_algorithm>`

Example:

We want to create keys to log on to `127.0.0.1` with `rsa` algorithm.

Using **ssh-keygen** : `ssh-keygen -t rsa`



This will create 2 keys (public/private) under `~/.ssh/` directory.

We'll change the permission of the `.ssh/` directory using `chmod`

```
sudo chmod -R 700 ~/.ssh/
```

Note: -R means recursively to change permission for all files and directories under `.ssh/`

After we create the keys we need to put the public key in the server side to make the server recognize us. There are many ways to do this but we will use `scp`.

SCP

It stands for Secure CoPy

To copy from server to client :

```
scp <user_name@host_name/ip:path_in_server> <path_in_client>
```

To copy from client to server :

```
scp <path_in_client> <user_name@server_name/ip:path_in_server>
```

Note: we need to copy the public key to the server and put it in `.ssh/authorized_keys` file. If it doesn't exist, create it.



After that we will go to `/etc/ssh/sshd_config` and change 3 things to **no** :

1. `challengeresponseauthentication` no
2. `passwordauthentication` no
3. `usepam` no

Then restart ssh service and it work fine.

```
sudo service ssh restart
```

Network Configuration

In this section we will know some network commands to configure a network and some network concepts.

ifconfig

It stands for InterFace CONFIGurator

This command will give you some useful information about a network.

like: your internal ip address (you can assign a new ip address with this command too), mac address, MTU(Maximum Transmission Unit) size and also you can enable or disable a network.

When you type `ifconfig`

It will provide you with 3 interfaces information.



eth0 -> This for wired network

lo -> This for the internal device network. [For more about lo](#)

wlan 0 -> This for wifi network

Let's take a close look in **wlan0** interface :



- **UP** -> it's enabled
- **BROADCAST** -> it supports broadcasting
- **RUNNING** -> it's operating
- **MULTICAST** -> it supports multicasting
- **MTU** -> the size of transmission unit (frame/packet)
- **INET** -> the local network ip
- **NETMASK** -> the netmask for the network. [for more info](#)
- **BROADCAST** -> the broadcast address

Assigning IP

```
ifconfig <interface_name> <new_ip> netmask <netmask_address>
```



Note: this will last until you close or reboot the system

***To active or inactive interface :** `* sudo ifconfig <interface_name> up/down`



Ping

This command tests the connectivity between 2 hosts.

We can use `ping` with `-c` option to specify the number of package will be sent. `ping -c 5 www.google.com`



Traceroute

This command show you the road to reach the host and the number of hops it passes through.

```
traceroute www.google.com
```



NSLOOKUP

This command searches for the ip of the given name

```
nslookup www.google.com
```



MTR

This command combines the functionality of the traceroute and ping programs in a single network diagnostic tool.

```
mtr www.google.com
```



Hostname

This command is for showing and changing the hostname

To show the hostname : `hostname`

To change the hostname: `hostname <new name>`

Note: this name will last until you close or reboot the system. To make it permanently you should write the new name in `/etc/hosts` and `/etc/hostname` files.

NMAP

This command is port scanner

Let's scan google ports : `nmap www.google.com`



Important Note: every command has options, try to have fun with these options and search for them in man pages.

Network Configuration File

In linux there are many important files to configure a network some of them are :

/etc/hosts

It has ips of the local hosts



/etc/protocols

It has protocols and their usage



/etc/services

It has tcp/udp services and their ports



/etc/resolve.conf

It has the ips of DNS servers



/etc/NetworkManager/system-connections/

This directory has all information about network you have logged in before.

