git clone https://github.com/Ashish-Garg524/LSS

Program-2

here we have 2 system with

system1: ip-162.10.10.10 and system2: ip-162.10.10.20

sudo apt-get install ftp

sudo apt-get install proftpd

sudo ifconfig

sudo systemctl stop NetworkManager

sudo ifconfig eno1 162.10.10.10 netmask 255.255.0.0 up

sudo ifconfig

ping 162.10.10.20

using put method:

pwd

cd /home/mca

ftp 162.10.10.20

user name: cna1

passwd: mca

ls

put <filename>

bye

ls

using get method:

pwd

cd /home/mca

ftp 162.10.10.20

user name: cna1

passwd: mca

ls

get <filename>

bye

ls

note: do same for both lab partners

program:3

here we have 2 system with

system1: ip-162.10.10.10 and system2: ip-162.10.10.20

for client :

sudo apt-get install openssh-client

sudo ifconfig

sudo systemctl stop NetworkManager

sudo ifconfig eno1 162.10.10.10 netmask 255.255.0.0 up

sudo ifconfig

ping 162.10.10.20

ssh-keygen

Generating public/private rsa key pair.

Enter file in which to save the key (/home/mca/.ssh/id\_rsa): home

Enter passphrase (empty for no passphrase):

Enter same passphrase again:

Your identification has been saved in home

Your public key has been saved in home.pub

ls

pwd

scp /home/mca/home.pub [cna1@162.10.10.20](mailto:cna1@162.10.10.20):/home/cna1

NOW SIMULTANEOUSLY AT SERVER SIDE on home screen

sudo apt-get install openssh-server

sudo ifconfig

sudo systemctl stop NetworkManager

sudo ifconfig eno1 162.10.10.20 netmask 255.255.0.0 up

sudo ifconfig

ping 162.10.10.10

ls

cd .ssh

ls

touch authorized\_keys

pwd

cd ..

cat home.pub >> *home/*cna1/.ssh/authorized\_keys

cd .ssh

cat authorized\_keys (to check if key moved or not)

cd ..

ls

NOW AT CLIENT SIDE DO

ssh -i /home/cna1/home [cna1@162.10.10.20](mailto:cna1@162.10.10.20) (to login)

passwd: mca

ls

touch authorized\_keys

Note:

1) home/cna1/home, the second home reffer the private key of client.

2) So what is happening is we sent our public key to server and server saved that public key to its authorized\_keys file under ssh directory.

3) Now when client login he will login using his private key which is saved as home inside home directory.

4) Also ( ~ ) is used when we are entering inside a directory. Here we can see we can do without using ~ sign also.

5) when client login inside server, the server password should not be asked at client side. If it ask, program execution failed. That is why use private key while login inside server using ssh.

Program:4

FOR THIS PROGRAM WE NOT NEED SWITCH, WE CAN DO THIS INDIVIDUALLY ON OUR OWN SYSTEM.

sudo apt-get install ipcalc

sudo ifconfig

sudo systemctl stop NetworkManager

sudo ifconfig eno1 162.10.10.10 netmask 255.255.0.0 up

sudo ifconfig

ping 162.10.10.10

The above step are not mandatory. Can be done without stopping network manage also.

ipcalc 11.10.10.20 (classA)

ipcalc 162.10.10.20 (classB)

ipcalc 191.10.10.20 (classC)

ipcalc <ip>/<subnet> -s <no. of hosts>

ipcalc 162.10.10.20/24 -s 16

note:

1) you have to show manual calculation also

program5:

AT SERVER SIDE

here we have 2 system with

system1: ip-162.10.10.10 and system2: ip-162.10.10.20

sudo apt-get install dnsmasq

sudo ifconfig

sudo systemctl stop NetworkManager

sudo ifconfig eno1 162.10.10.10 netmask 255.255.0.0 up

sudo ifconfig

ping 162.10.10.20

sudo systemctl stop dnsmasq(check for INACTIVE)

sudo lsof -i:53 (check for systemd-resolved services)

sudo nano /etc/dnsmasq.conf

port=53

bogus-priv

strict-order

expand-hosts

dhcp-range=162.10.10.1,162.10.10.200

dhcp-option=option:router,162.10.10.1

dhcp-option=option:netmask,255.255.0.0

sudo systemctl stop systemd-resolved

sudo systemctl start dnsmasq

sudo systemctl status dnsmasq( it should show ACTIVE)

AT CLIENT SIDE

sudo dhclient -v

METHOD-2

AT SERVER SIDE

sudo apt-get install dnsmasq

sudo ifconfig

sudo systemctl stop NetworkManager

sudo ifconfig eno1 162.10.10.10 netmask 255.255.0.0 up

sudo ifconfig

ping 162.10.10.20

sudo systemctl stop dnsmasq

sudo systemctl status dnsmasq(check for INACTIVE)

sudo lsof -i:53 (check for systemd-resolved services)

sudo dnsmasq -i eno1 -I lo -a 162.10.10.10 --dhcp-range=162.10.10.1,162.10.10.50 -d -u root --log-dhcp -q

AT CLIENT SIDE

sudo dhclient -v

NOTE:if above command do not work, stop systemd-resolved services and then run the command.

program 6:

sudo apt-get install dnsmasq

sudo apt-get install dnsutils

sudo ifconfig

sudo systemctl stop NetworkManager

sudo ifconfig eno1 162.10.10.10 netmask 255.255.0.0 up

sudo ifconfig

ping 162.10.10.10

sudo nano /etc/dnsmasq.conf

#DNS CONFIGURATION

port=53

domain-needed

bogus-priv

\ strict-order

expand-hosts

domain=ashish.in

sudo nano /etc/hosts

192.168.10.20 ashish.in

sudo systemctl stop systemd-resolved

sudo systemctl start dnsmasq

sudo systemctl status dnsmasq

sudo dig a ashish.in @localhost

program:7

AT ONE SYSTEM

here we have 2 system with

system1: ip-162.10.10.10 and system2: ip-162.10.10.20

sudo apt-get install iptables

sudo ifconfig

sudo systemctl stop NetworkManager

sudo ifconfig eno1 162.10.10.10 netmask 255.255.0.0 up

sudo ifconfig

ping 162.10.10.20

AT ANOTHER SYSTEM

sudo iptables -A INPUT -s 162.10.10.20 -j DROP

sudo iptables -L

sudo iptables -F

sudo iptables -A INPUT -m iprange --src-range 162.10.10.5-162.10.10.30 -j REJECT

sudo iptables -L

sudo iptables -F

sudo iptables -A INPUT -s 162.10.10.20 -j DROP

sudo iptables -A INPUT -s 162.10.10.20 -j ACCEPT

program8

sudo apt-get install net-tools

sudo apt-get install traceroute

sudo ifconfig

sudo systemctl stop NetworkManager

sudo ifconfig eno1 162.10.10.10 netmask 255.255.0.0 up

sudo ifconfig

ping 162.10.10.20

ping 162.10.10.10

traceroute 162.10.10.10

sudo ifconfig ethernetid

nslookup <ip>

netstat -a

route <ip>

program:9

sudo apt-get install netcat

sudo ifconfig

sudo systemctl stop NetworkManager

sudo ifconfig eno1 162.10.10.10 netmask 255.255.0.0 up

sudo ifconfig

ping 162.10.10.20

AS server

nc -lv <port no.> eg1234

as client

nc -v <ip of server> <port no.>

program10:

here we have 2 system with

system1: ip-162.10.10.10 and system2: ip-162.10.10.20

at server side:

sudo apt-get install squid

sudo apt-get install lynx

sudo apt install python3

sudo ifconfig

sudo systemctl stop NetworkManager

sudo ifconfig eno1 162.10.10.10 netmask 255.255.0.0 up

sudo ifconfig

ping 162.10.10.20

mkdir /tmp/website

cd /tmp/website

echo "hello mca001" > index.html

sudo apt install python3

python3 -m http.server 5000 --bind 127.0.0.30

Run [http://127.0.0.30:5000](http://127.0.0.30:5000/) in web browser at server side to check if if showing result or not.

At client side:

sudo ifconfig

sudo systemctl stop NetworkManager

sudo ifconfig eno1 162.10.10.20 netmask 255.255.0.0 up

sudo ifconfig

ping 162.10.10.10

ssh [cna1@162.10.10.10](mailto:cna1@162.10.10.10) -v -L 127.0.0.20:8000:127.0.0.30.5000

Run [http://127.0.0.20:8000](http://127.0.0.30:5000/) in web browser at client side

Note :

1) ssh [cna1@162.10.10.10](mailto:cna1@162.10.10.10) -v -L 127.0.0.20:8000:127.0.0.30.5000

in this cna1 is server user name and 162.10.10.10 is also server ip address.