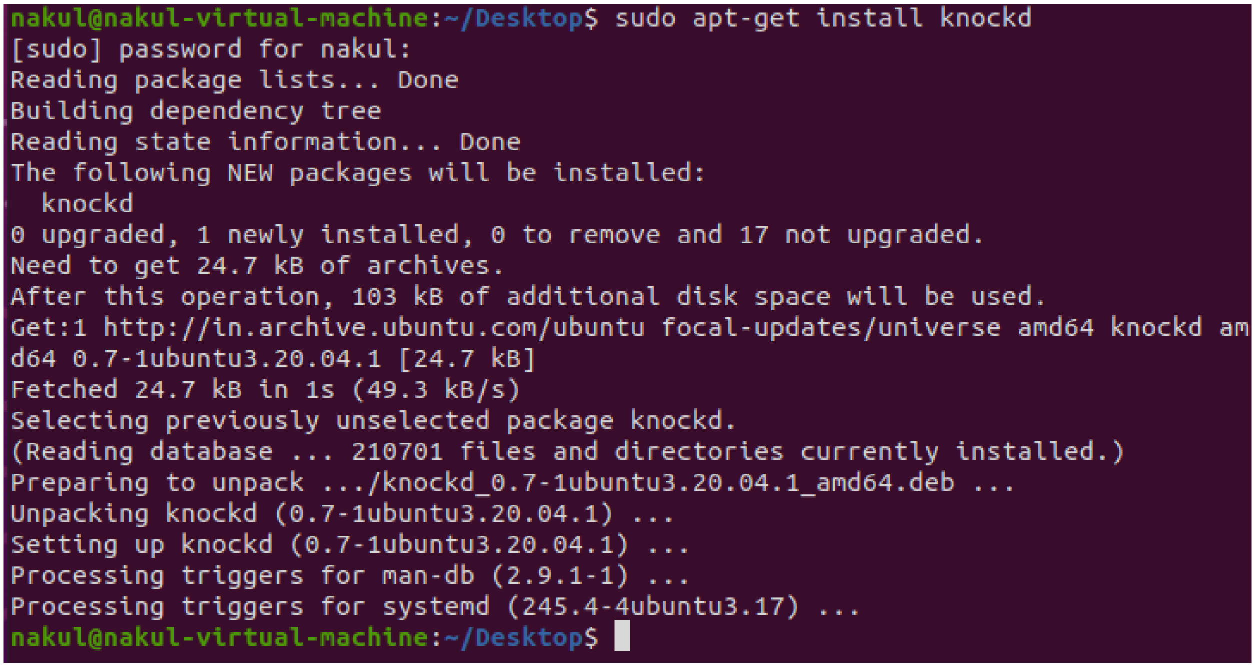
Fundamentals of Computer Security
(CSE 345/CSE 545)
Assignment 1
Question 4

Nakul Thureja
2020528

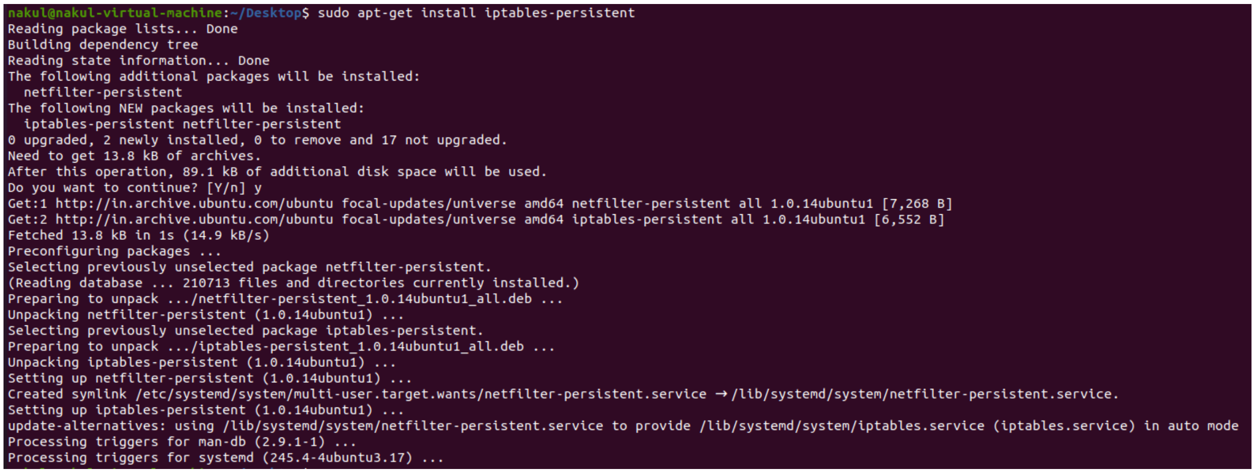
Part A

Step 1: Installing the required packages

1. knockd
2. iptables-persistent



```
nakul@nakul-virtual-machine:~/Desktop$ sudo apt-get install knockd
[sudo] password for nakul:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  knockd
0 upgraded, 1 newly installed, 0 to remove and 17 not upgraded.
Need to get 24.7 kB of archives.
After this operation, 103 kB of additional disk space will be used.
Get:1 http://in.archive.ubuntu.com/ubuntu focal-updates/universe amd64 knockd am
d64 0.7-1ubuntu3.20.04.1 [24.7 kB]
Fetched 24.7 kB in 1s (49.3 kB/s)
Selecting previously unselected package knockd.
(Reading database ... 210701 files and directories currently installed.)
Preparing to unpack .../knockd_0.7-1ubuntu3.20.04.1_amd64.deb ...
Unpacking knockd (0.7-1ubuntu3.20.04.1) ...
Setting up knockd (0.7-1ubuntu3.20.04.1) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for systemd (245.4-4ubuntu3.17) ...
nakul@nakul-virtual-machine:~/Desktop$ 
```



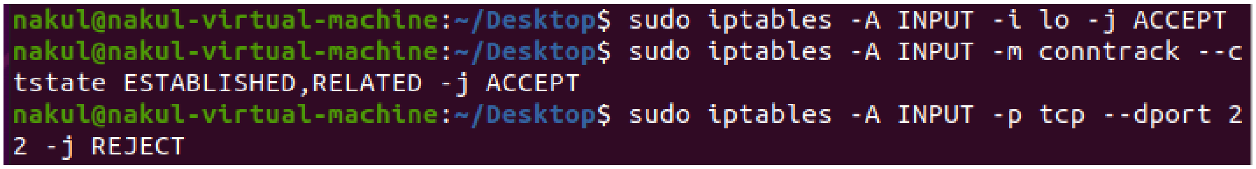
```
nakul@nakul-virtual-machine:~/Desktop$ sudo apt-get install iptables-persistent
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  netfilter-persistent
The following NEW packages will be installed:
  iptables-persistent netfilter-persistent
0 upgraded, 2 newly installed, 0 to remove and 17 not upgraded.
Need to get 13.8 kB of archives.
After this operation, 89.1 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://in.archive.ubuntu.com/ubuntu focal-updates/universe amd64 netfilter-persistent all 1.0.14ubuntu1 [7,268 B]
Get:2 http://in.archive.ubuntu.com/ubuntu focal-updates/universe amd64 iptables-persistent all 1.0.14ubuntu1 [6,552 B]
Fetched 13.8 kB in 1s (14.9 kB/s)
Preconfiguring packages ...
Selecting previously unselected package netfilter-persistent.
(Reading database ... 210713 files and directories currently installed.)
Preparing to unpack .../netfilter-persistent_1.0.14ubuntu1_all.deb ...
Unpacking netfilter-persistent (1.0.14ubuntu1) ...
Selecting previously unselected package iptables-persistent.
Preparing to unpack .../iptables-persistent_1.0.14ubuntu1_all.deb ...
Unpacking iptables-persistent (1.0.14ubuntu1) ...
Setting up netfilter-persistent (1.0.14ubuntu1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/netfilter-persistent.service → /lib/systemd/system/netfilter-persistent.service.
Setting up iptables-persistent (1.0.14ubuntu1) ...
update-alternatives: using /lib/systemd/system/netfilter-persistent.service to provide /lib/systemd/system/iptables.service (iptables.service) in auto mode
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for systemd (245.4-4ubuntu3.17) ...
```

Step 2:

The command “sudo iptables -A INPUT -i lo -j Accept” makes sure that the data will not be blocked from my local machine as lo here refers to the local network.

The command “sudo iptables -A INPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT” makes sure that the current ssh connection is not blocked.

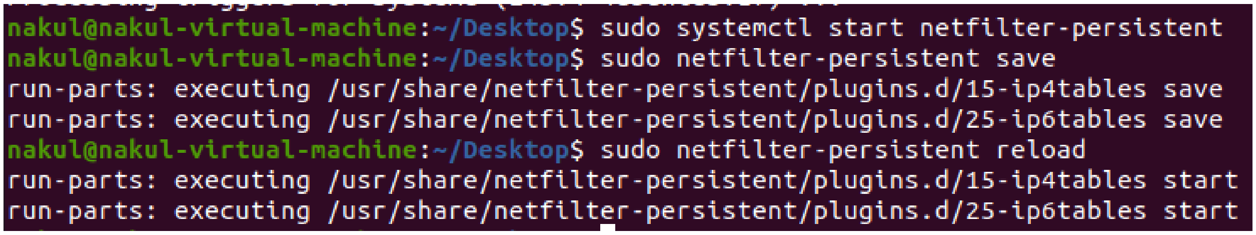
The command “sudo iptables -A INPUT -p tcp --dport 22 -j REJECT” drops the packets at port 22 by the firewall.



```
nakul@nakul-virtual-machine:~/Desktop$ sudo iptables -A INPUT -i lo -j ACCEPT
nakul@nakul-virtual-machine:~/Desktop$ sudo iptables -A INPUT -m conntrack --c
tstate ESTABLISHED,RELATED -j ACCEPT
nakul@nakul-virtual-machine:~/Desktop$ sudo iptables -A INPUT -p tcp --dport 2
2 -j REJECT
```

Step 3:

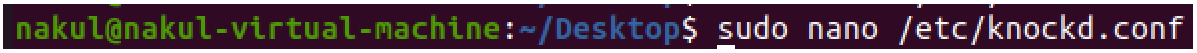
Start the netfilter-persistent services, to update the iptable rules



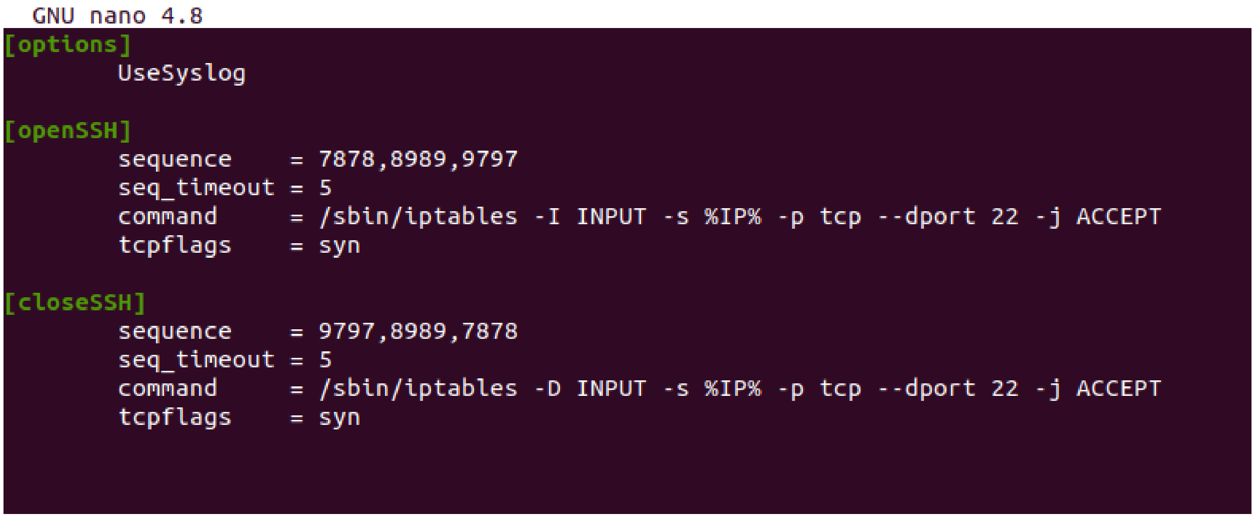
```
nakul@nakul-virtual-machine:~/Desktop$ sudo systemctl start netfilter-persistent
nakul@nakul-virtual-machine:~/Desktop$ sudo netfilter-persistent save
run-parts: executing /usr/share/netfilter-persistent/plugins.d/15-ip4tables save
run-parts: executing /usr/share/netfilter-persistent/plugins.d/25-ip6tables save
nakul@nakul-virtual-machine:~/Desktop$ sudo netfilter-persistent reload
run-parts: executing /usr/share/netfilter-persistent/plugins.d/15-ip4tables start
run-parts: executing /usr/share/netfilter-persistent/plugins.d/25-ip6tables start
```

Step 4:

Edit the knockd.conf file to change the sequence of ports and to make sure the rules are inserted at the top of the iptables using ‘-I’ instead of ‘-A’

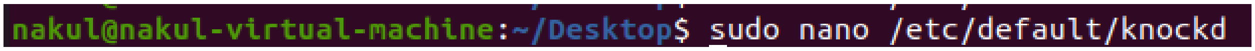


```
nakul@nakul-virtual-machine:~/Desktop$ sudo nano /etc/knockd.conf
```

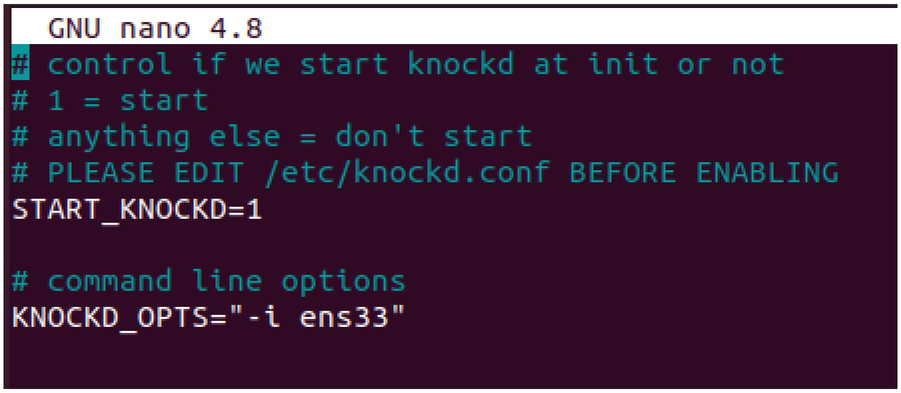


```
  GNU nano 4.8
[options]
        UseSyslog

[openSSH]
        sequence    = 7878,8989,9797
        seq_timeout = 5
        command     = /sbin/iptables -I INPUT -s %IP% -p tcp --dport 22 -j ACCEPT
        tcpflags    = syn

[closeSSH]
        sequence    = 9797,8989,7878
        seq_timeout = 5
        command     = /sbin/iptables -D INPUT -s %IP% -p tcp --dport 22 -j ACCEPT
        tcpflags    = syn
```

Step 5:
Edit the default knockd file to start the the Knockd and change the network name to ens33 which corresponds to the virtual machine used.

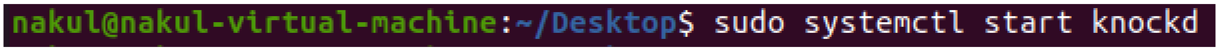


```
nakul@nakul-virtual-machine:~/Desktop$ sudo nano /etc/default/knockd
```



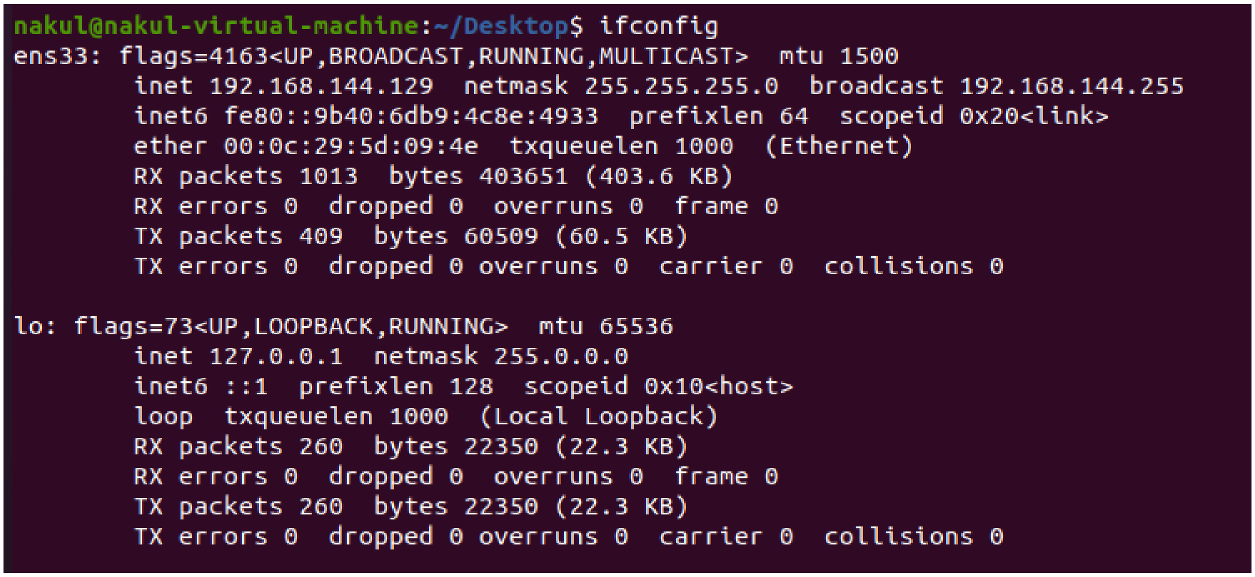
```
  GNU nano 4.8
# control if we start knockd at init or not
# 1 = start
# anything else = don't start
# PLEASE EDIT /etc/knockd.conf BEFORE ENABLING
START_KNOCKD=1

# command line options
KNOCKD_OPTS="-i ens33"
```

Step 6:
The command “sudo systemctl start knockd” starts the service



```
nakul@nakul-virtual-machine:~/Desktop$ sudo systemctl start knockd
```

Step 7:
The command “ifconfig” returns the ip address of the VM to be used for ssh connection. Which is 192.168.144.129



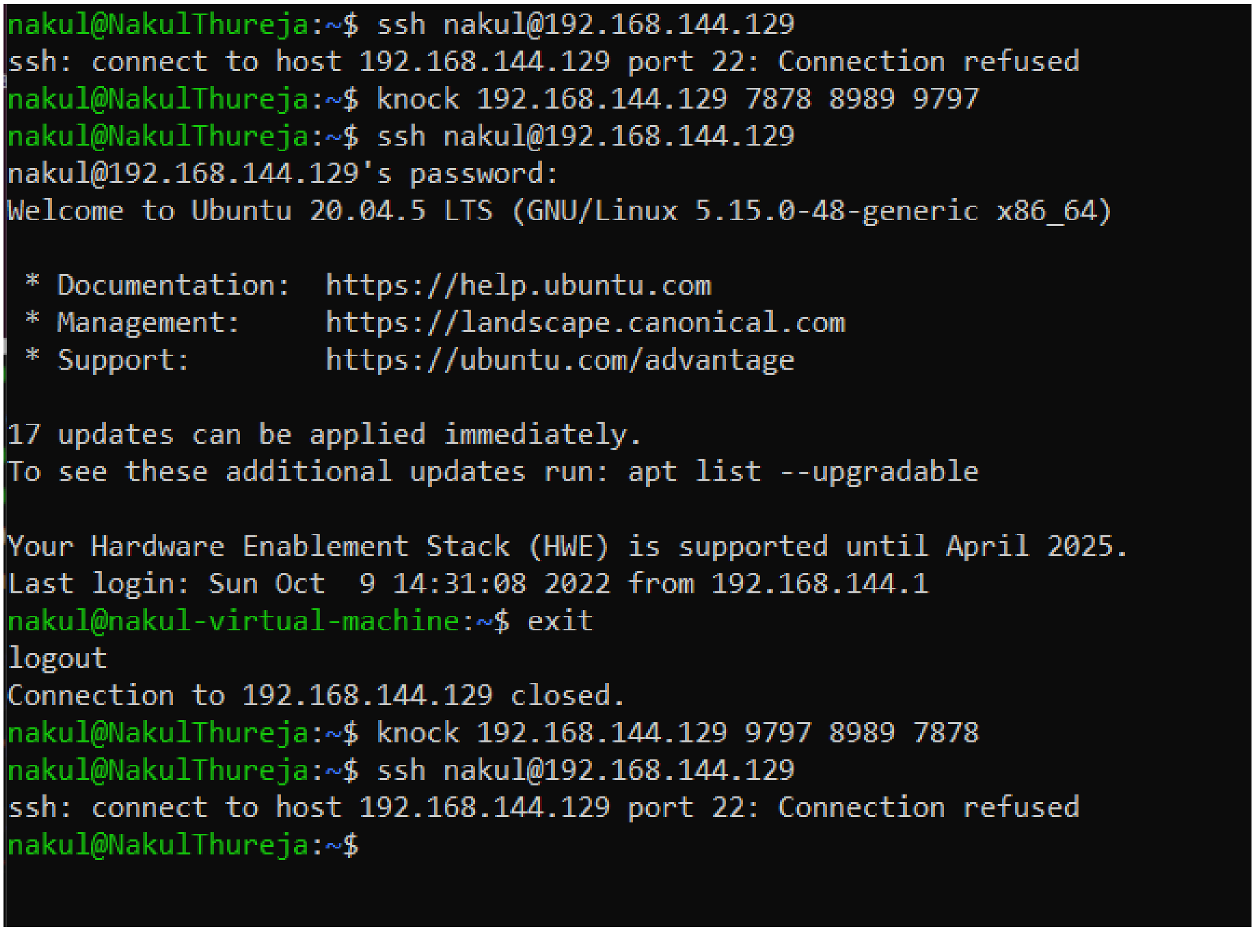
```
nakul@nakul-virtual-machine:~/Desktop$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.144.129  netmask 255.255.255.0  broadcast 192.168.144.255
        inet6 fe80::9b40:6db9:4c8e:4933  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:5d:09:4e  txqueuelen 1000  (Ethernet)
        RX packets 1013  bytes 403651 (403.6 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 409  bytes 60509 (60.5 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 260  bytes 22350 (22.3 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 260  bytes 22350 (22.3 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Connecting the ssh:

I have used wsl (windows subsystem for linux) to connect to ssh. The command “ssh nakul@192.168.144.129”

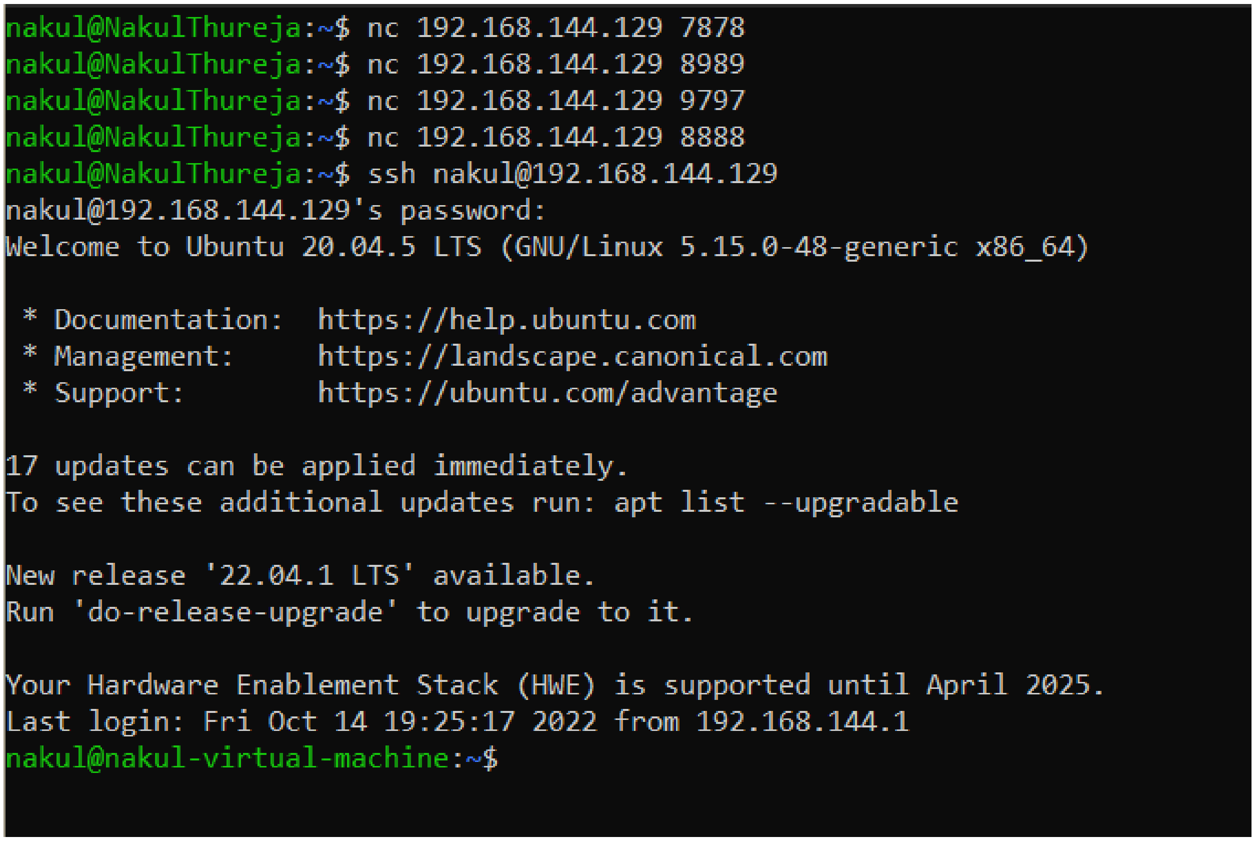
Initially the connection the refused but when we use the knock command to knock the IP in a particular connection it allows us to connect to th VM via ssh connection.

To close the connection we knock in the order specified to close the connection after which requests for ssh will not be accepted by the VM.



```
nakul@NakulThureja:~$ ssh nakul@192.168.144.129
ssh: connect to host 192.168.144.129 port 22: Connection refused
nakul@NakulThureja:~$ knock 192.168.144.129 7878 8989 9797
nakul@NakulThureja:~$ ssh nakul@192.168.144.129
nakul@192.168.144.129's password:
Welcome to Ubuntu 20.04.5 LTS (GNU/Linux 5.15.0-48-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

17 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Your Hardware Enablement Stack (HWE) is supported until April 2025.
Last login: Sun Oct  9 14:31:08 2022 from 192.168.144.1
nakul@nakul-virtual-machine:~$ exit
logout
Connection to 192.168.144.129 closed.
nakul@NakulThureja:~$ knock 192.168.144.129 9797 8989 7878
nakul@NakulThureja:~$ ssh nakul@192.168.144.129
ssh: connect to host 192.168.144.129 port 22: Connection refused
nakul@NakulThureja:~$
```

But knockd is not the only way to connect. The nc (netcat) command can also be used to connect to ports in a certain order to give access to ssh connections of the VM.



```
nakul@NakulThureja:~$ nc 192.168.144.129 7878
nakul@NakulThureja:~$ nc 192.168.144.129 8989
nakul@NakulThureja:~$ nc 192.168.144.129 9797
nakul@NakulThureja:~$ nc 192.168.144.129 8888
nakul@NakulThureja:~$ ssh nakul@192.168.144.129
nakul@192.168.144.129's password:
Welcome to Ubuntu 20.04.5 LTS (GNU/Linux 5.15.0-48-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

17 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

New release '22.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Your Hardware Enablement Stack (HWE) is supported until April 2025.
Last login: Fri Oct 14 19:25:17 2022 from 192.168.144.1
nakul@nakul-virtual-machine:~$
```

Part B

TCP is preferred over UDP as TCP is a connection oriented protocol compared to UDP which is a Connectionless Protocol.
Tcp requires proper closing of connection as compared to UDP where there is no strict requirement of closing the connections.
Also TCP sends acknowledgement for the packets which makes sure that the data is received properly.
Lastly TCP is an in order protocol comnpared the UDP which is an out of order protocol.
Therefore, TCP is clearly a better choice for ssh port knocking.

Part C

The default configuration of knock is 7000, 8000, 9000. This configuration is obviously not secure as everyone knows this and this might be an attackers first guess while trying to access a machine. Even though it requires a password to break into the VM it stills gives the attacker a platform to try and crack the password using password breaking tools or brute forcing, so it is always a better choice to change the default ports.

Keeping any other ports rather than the default gives an extra layer of protection to the user but its still not completely secure.

Better option will be to have random port knocking sequence using time stamps or some other form of dynamic sequence which only a user can enter.