

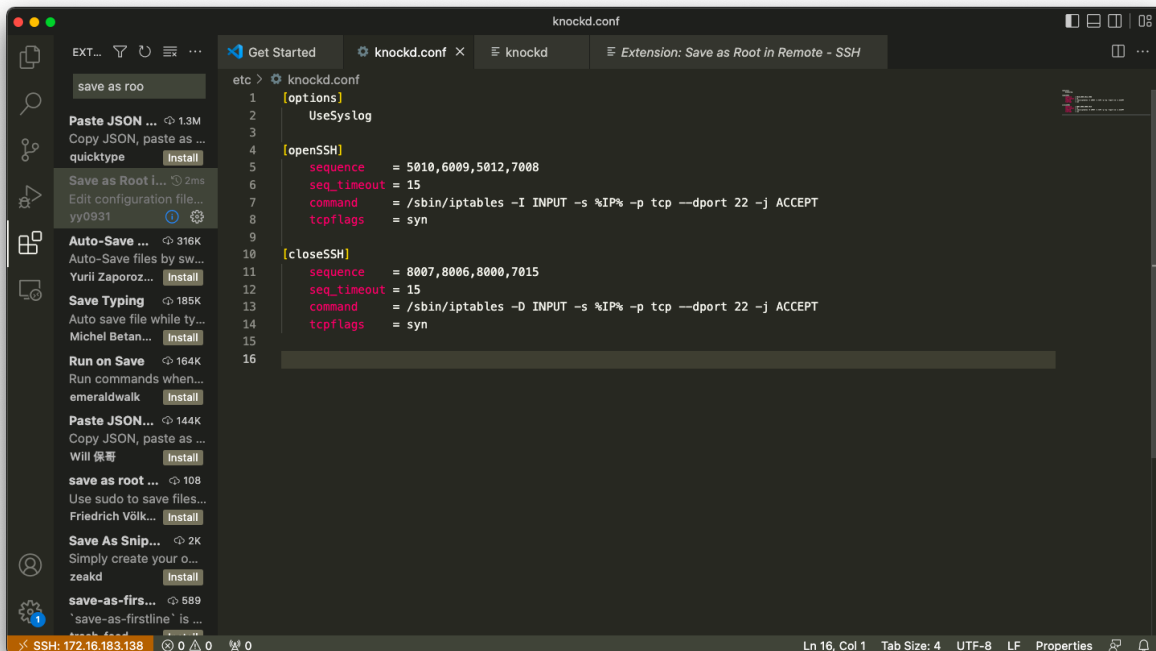
FCS ASSIGNMENT 1

Report for Question 4

Answer for part a:

Steps to configure knockd for port 22 on a VM:

- Installation of a new Ubuntu Server: I installed a new VM server from [here](#). I set it up using ISO image in VMWare Fusion. The IP addresses of the 2 servers I setup were: 172.16.183.138 and 172.16.183.138.
- Installation of knockd: On the server I installed 'knockd' using the command `sudo apt install knockd`. This would act as the machine whose SSH access I would like to restrict using knockd. On my MAC laptop, I installed 'knock' client. This would act as a client that wishes to get SSH access on the server.
- Then I verified if there were any IP TABLE rules present in the VM using command: `sudo iptables -L`. There were no pre-existing IP Table rules. If there were, I would have removed them.
- Then I checked whether the firewall for the VM was active using command: `sudo ufw status`. It was inactive, I activated that using: `sudo ufw enable`.
- After that I modified the file `/etc/knockd.conf`. File after modification:



```
etc > knockd.conf
1 [options]
2   UseSyslog
3
4 [openSSH]
5   sequence    = 5010,6009,5012,7008
6   seq_timeout = 15
7   command     = /sbin/iptables -I INPUT -s %IP% -p tcp --dport 22 -j ACCEPT
8   tcpflags    = syn
9
10
11 [closeSSH]
12   sequence    = 8007,8006,8000,7015
13   seq_timeout = 15
14   command     = /sbin/iptables -D INPUT -s %IP% -p tcp --dport 22 -j ACCEPT
15   tcpflags    = syn
16
```

For allowing SSH connection, [openSSH] was modified . I set up the port knocking sequence 5010, 6009, 5012, 7008 for allowing the client to get SSH access. The ports

should be knocked within 15 seconds with the sequence, as mentioned in `seq_timeout`. If the sequence is not completed within 15 seconds, the attempt is disregarded. The command field specifies the command to be executed when the ports are knocked according to the sequence.

Explanation of command field:

In the command field, `/sbin/iptables` tells the location where the command needs to be written.

`-I` flag specifies that this command must be inserted as the first line of the IP table rules, so that it is not overridden by any other command.

`INPUT` field specifies that the command needs to be inserted in `INPUT` chain of the IP table

`-s %IP%` specifies the source IP. `%IP%` gets replaced with the knocker's IP.

`--dport 22` specifies that the destination port is port 22 which is the default port for SSH.

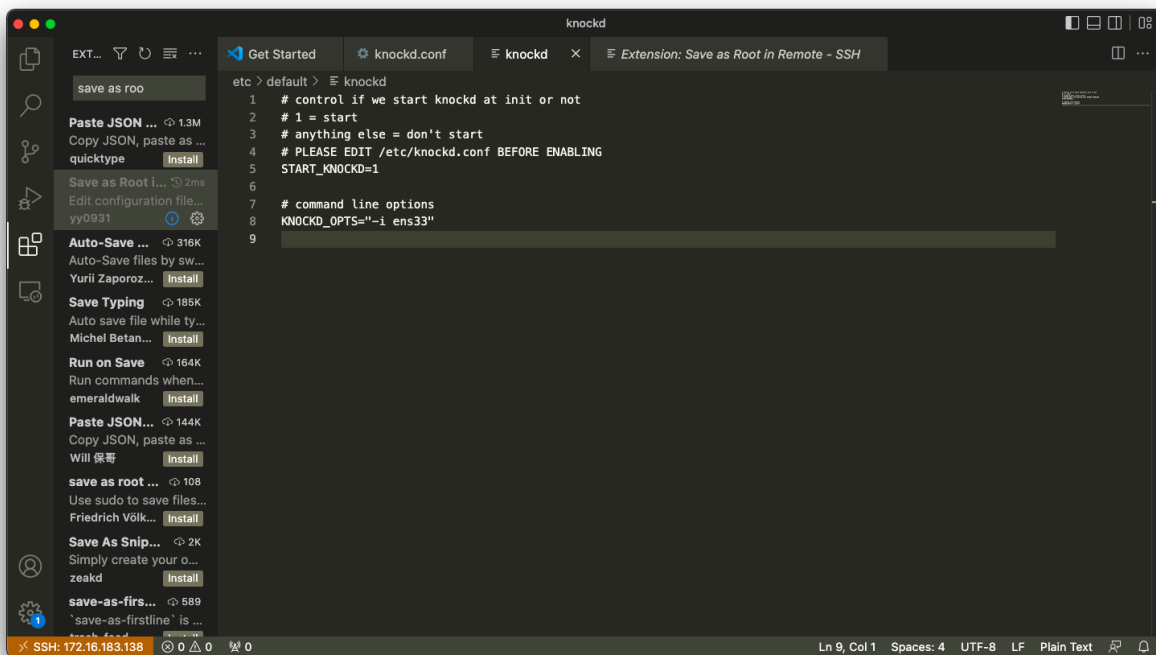
`-j ACCEPT` specifies the action of accepting the packet should be performed.

`tcpflags = syn` is used to ensure that the packets having `syn` flag set are only considered for knocking.

- Similar things are mentioned in [closeSSH]. After knocking the ports sequence mentioned for closing the SSH, the client will not be able to set up a new SSH connection on the server. If the client already has an SSH connection, that won't be disturbed. For the purpose of closing the SSH port, I have put the sequence as 8007, 8006, 8000, 7015. As before, it must be knocked with the sequence within a span of 15 seconds.

Explanation of command: From the case of 'opening' the SSH port for connections, the only difference is that I have used `-D` instead of `-I`. This is to delete the iptable rule from the INPUT chain so that there are no more connections to SSH.

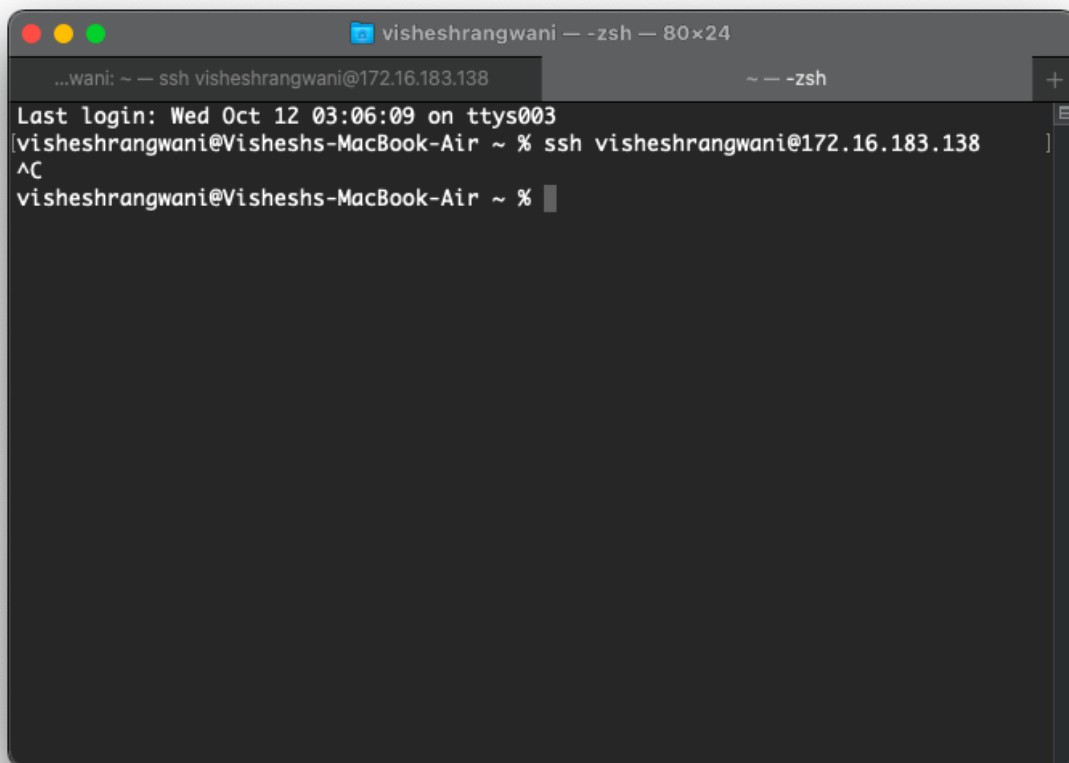
- After modifying this `knockd.conf` file, the `knockd` daemon had to be enabled. This was done via the file: `/etc/default/knockd`. I had enabled the daemon so that `knockd` is always on. This was done by setting the value of `START_KNOCD=1`. Apart from this the interface `ens33`, as found from `ip addr` command was changed and the line was uncommented. The file after modification looks like this:



- After doing this, knockd service was started in the VM using the following commands:

```
sudo systemctl start knockd
```

```
sudo systemctl enable knockd
```
- After setting up knockd in VM, I tried to get SSH access from the client (MAC laptop).
Before knocking, I was unable to get SSH access:

A terminal window titled "visheshrangwani — zsh — 80x24" with standard macOS window controls. The terminal shows an SSH session from a local machine to 172.16.183.138. The prompt is "...wani: ~ — ssh visheshrangwani@172.16.183.138". The output shows the last login time and the user's command to run 'ssh' again, which is interrupted by a carriage return (^C).

```
visheshrangwani — zsh — 80x24
...wani: ~ — ssh visheshrangwani@172.16.183.138
Last login: Wed Oct 12 03:06:09 on ttys003
visheshrangwani@Visheshs-MacBook-Air ~ % ssh visheshrangwani@172.16.183.138
^C
visheshrangwani@Visheshs-MacBook-Air ~ %
```

Then I tried to knock the ports using the command:

```
knock -v 172.16.183.138 5010 6009 5012 7008
```

Then the VM responded, asking for password:

```
visheshrangwani — ssh visheshrangwani@172.16.183.138 — 99x24
...@visheshrangwani: ~ — ssh visheshrangwani@172.16.183.138 ~ — ssh visheshrangwani@172.16.183.138
Last login: Wed Oct 12 03:06:09 on ttys003
visheshrangwani@Visheshs-MacBook-Air ~ % ssh visheshrangwani@172.16.183.138
visheshrangwani@Visheshs-MacBook-Air ~ % knock -v 172.16.183.138 5010 6009 5012 7008
hitting tcp 172.16.183.138:5010
hitting tcp 172.16.183.138:6009
hitting tcp 172.16.183.138:5012
hitting tcp 172.16.183.138:7008
visheshrangwani@Visheshs-MacBook-Air ~ % ssh visheshrangwani@172.16.183.138
visheshrangwani@172.16.183.138's password: ?
```

After entering the password, I was able to access the VM using SSH:

```
visheshrangwani — visheshrangwani@visheshrangwani: ~ — ssh visheshrangwani@172.16.183.138 — 176x43
~ — visheshrangwani@visheshrangwani: ~ — ssh visheshrangwani@172.16.183.138 ~ — visheshrangwani@visheshrangwani: ~ — ssh visheshrangwani@172.16.183.138
Last login: Wed Oct 12 03:06:09 on ttys003
visheshrangwani@Visheshs-MacBook-Air ~ % ssh visheshrangwani@172.16.183.138
visheshrangwani@Visheshs-MacBook-Air ~ % knock -v 172.16.183.138 5010 6009 5012 7008
hitting tcp 172.16.183.138:5010
hitting tcp 172.16.183.138:6009
hitting tcp 172.16.183.138:5012
hitting tcp 172.16.183.138:7008
visheshrangwani@Visheshs-MacBook-Air ~ % ssh visheshrangwani@172.16.183.138
visheshrangwani@172.16.183.138's password:
Welcome to Ubuntu 20.04.5 LTS (GNU/Linux 5.4.0-128-generic x86_64)

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

System information as of Tue 11 Oct 2022 10:17:20 PM UTC

System load:  0.14          Processes:    231
Usage of /:   32.7% of 13.67GB   Users logged in: 1
Memory usage: 27%             IPv4 address for ens33: 172.16.183.138
Swap usage:   0%

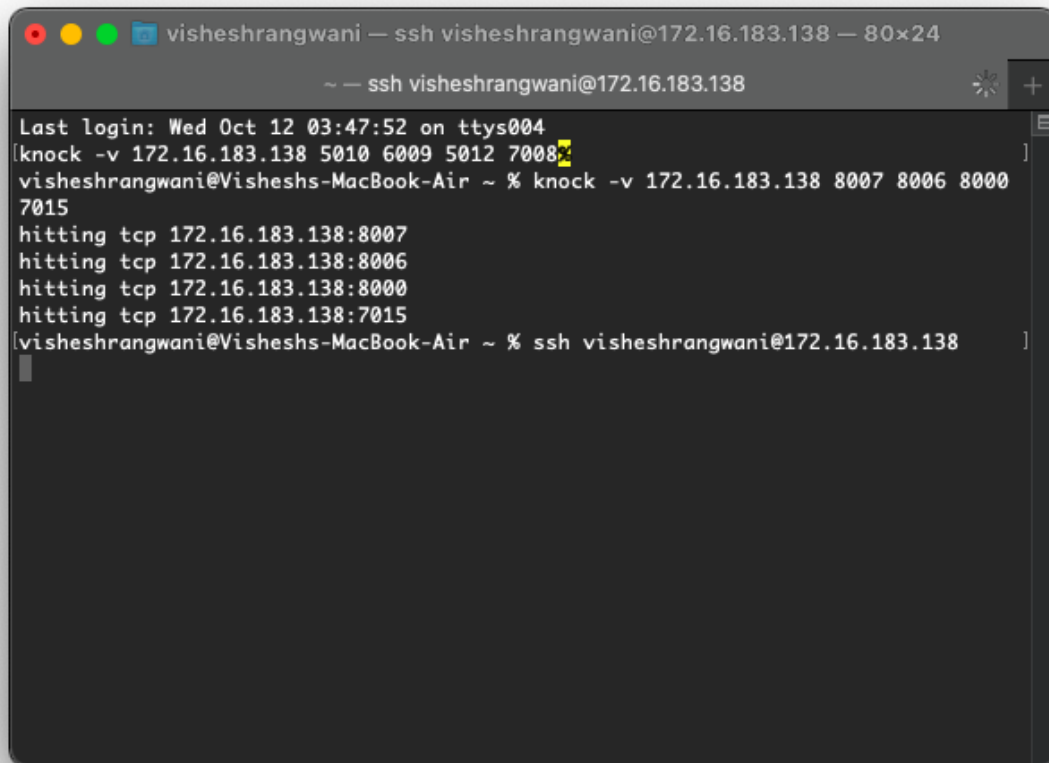
11 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

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

Last login: Tue Oct 11 21:38:11 2022 from 172.16.183.1
visheshrangwani@visheshrangwani: ~
```

Then I knocked the machine with the closing sequence and I could not access the VM using SSH.

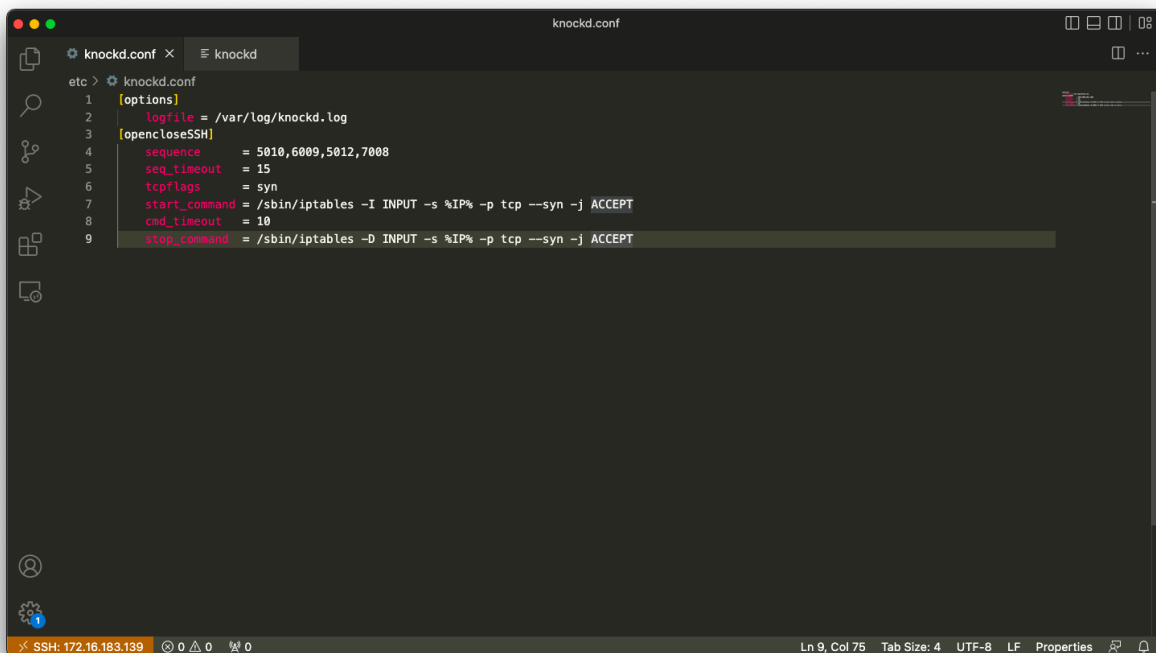
Command: `knock -v 172.16.183.138 8007 8006 8000 7015`



```
visheshrangwani — ssh visheshrangwani@172.16.183.138 — 80x24
~ — ssh visheshrangwani@172.16.183.138

Last login: Wed Oct 12 03:47:52 on ttys004
[knock -v 172.16.183.138 5010 6009 5012 7008]
visheshrangwani@Visheshs-MacBook-Air ~ % knock -v 172.16.183.138 8007 8006 8000
7015
hitting tcp 172.16.183.138:8007
hitting tcp 172.16.183.138:8006
hitting tcp 172.16.183.138:8000
hitting tcp 172.16.183.138:7015
[visheshrangwani@Visheshs-MacBook-Air ~ % ssh visheshrangwani@172.16.183.138]
```

- However there was a problem with this approach, once the opening sequence was entered, the access had to manually be restricted by knocking with the closing sequence. To overcome this problem and make the IP table rules more practical and usable, I made another VM server and its knockd.conf file was written with a different set of commands. Here is a screenshot of the modified knockd.conf file:



```
etc > knockd.conf
1 [options]
2 logfile = /var/log/knockd.log
3 [opencloseSSH]
4 sequence = 5010,6009,5012,7008
5 seq_timeout = 15
6 tcpflags = syn
7 start_command = /sbin/iptables -I INPUT -s %IP% -p tcp --syn -j ACCEPT
8 cmd_timeout = 10
9 stop_command = /sbin/iptables -D INPUT -s %IP% -p tcp --syn -j ACCEPT
```

SSH: 172.16.183.139 0 0 0 Ln 9, Col 75 Tab Size: 4 UTF-8 LF Properties

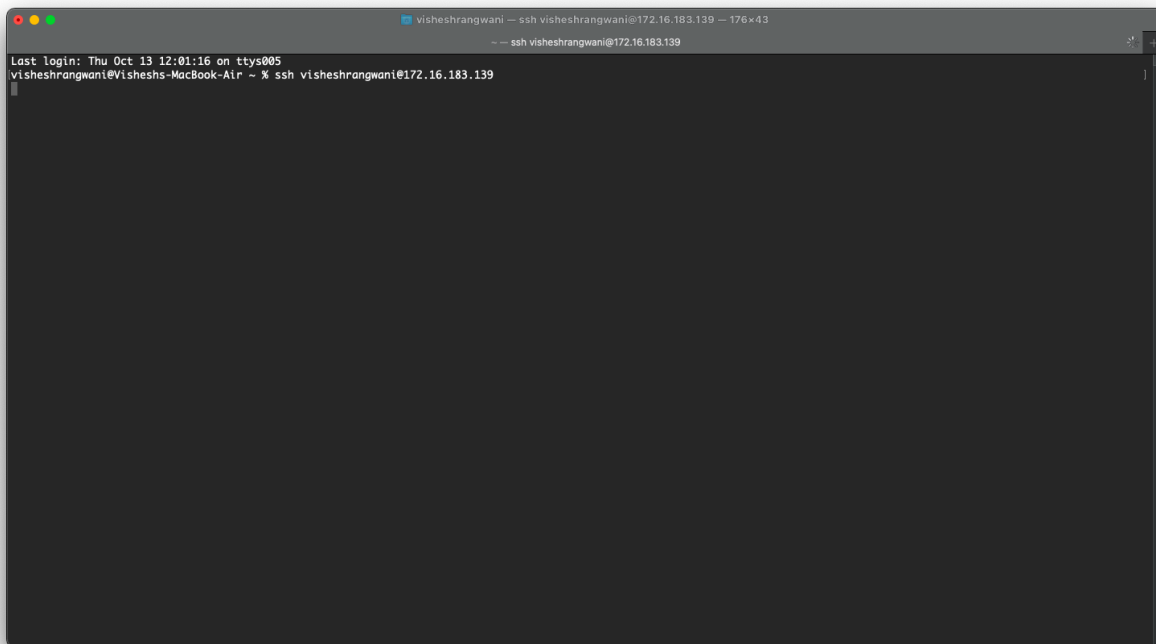
In this, there is `start_command` and `stop_command`. This ensures that both these are executed whenever the knocking port sequence is knocked by a client. The `start_command` allows the client to access the VM using SSH but after 10 seconds (`cmd_timeout`), the `stop_command` gets executed on its own and the SSH port would be closed automatically.

Using this the closing sequence will not have to be executed explicitly by the client. This ensures that there won't be a long time for which the SSH port would be open for connection and would not lead to too many connections when the port is opened. So anyone having password won't be able to access the VM using SSH unless he knows knocking sequence.

- The rest of the steps for this VM as well were the same as before and knockd was enabled.

In this new machine, the various scenario's screenshots are attached:

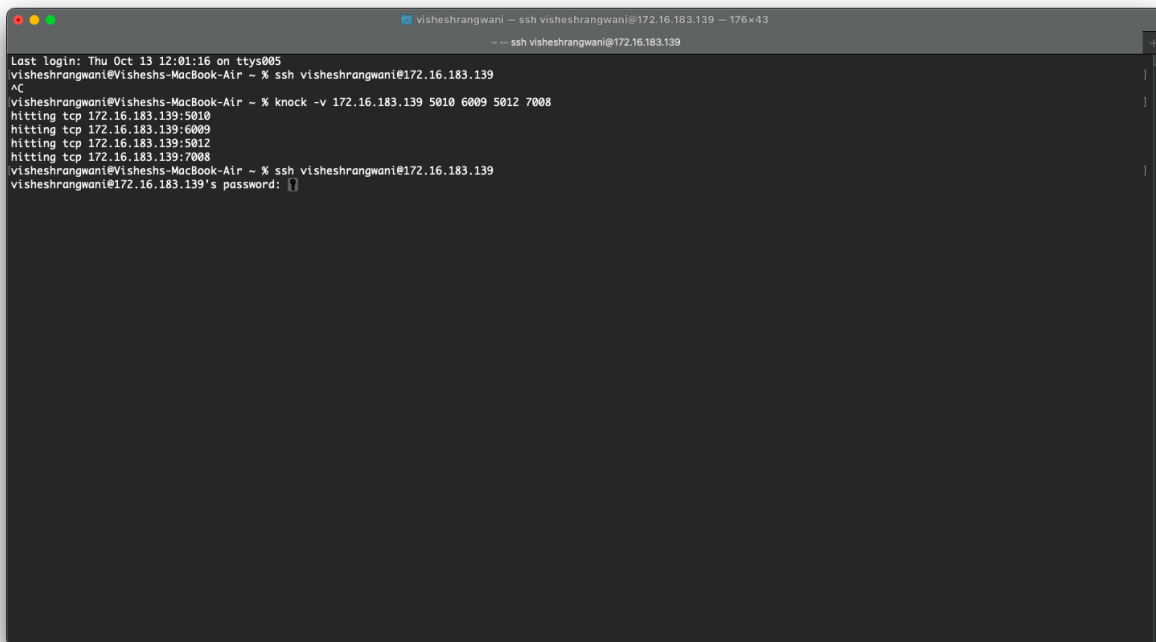
Trying without knocking:



```
visheshrangwani — ssh visheshrangwani@172.16.183.139 — 176x43
-- ssh visheshrangwani@172.16.183.139
Last login: Thu Oct 13 12:01:16 on ttys005
visheshrangwani@Visheshs-MacBook-Air ~ % ssh visheshrangwani@172.16.183.139
```

Unable to login using SSH as port is closed

After knocking:



```
visheshrangwani — ssh visheshrangwani@172.16.183.139 — 176x43
-- ssh visheshrangwani@172.16.183.139
Last login: Thu Oct 13 12:01:16 on ttys005
visheshrangwani@Visheshs-MacBook-Air ~ % ssh visheshrangwani@172.16.183.139
AC
visheshrangwani@Visheshs-MacBook-Air ~ % knock -v 172.16.183.139 5010 6009 5012 7008
hitting tcp 172.16.183.139:5010
hitting tcp 172.16.183.139:6009
hitting tcp 172.16.183.139:5012
hitting tcp 172.16.183.139:7008
visheshrangwani@Visheshs-MacBook-Air ~ % ssh visheshrangwani@172.16.183.139
visheshrangwani@172.16.183.139's password: 
```

Allows you to login after entering your password.


```
visheshrangwani — visheshrangwani@visheshrangwani: ~ — ssh visheshrangwani@172.16.183.139 — 176x43
— visheshrangwani@visheshrangwani: ~ — ssh visheshrangwani@172.16.183.139

Last login: Thu Oct 13 12:01:16 on ttys005
visheshrangwani@Visheshs-MacBook-Air ~ % ssh visheshrangwani@172.16.183.139
AC
visheshrangwani@Visheshs-MacBook-Air ~ % knock -v 172.16.183.139 5010 6009 5012 7008
hitting tcp 172.16.183.139:5010
hitting tcp 172.16.183.139:6009
hitting tcp 172.16.183.139:5012
hitting tcp 172.16.183.139:7008
visheshrangwani@Visheshs-MacBook-Air ~ % ssh visheshrangwani@172.16.183.139
visheshrangwani@172.16.183.139's password:
Welcome to Ubuntu 20.04.5 LTS (GNU/Linux 5.4.0-128-generic x86_64)

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

System information as of Thu 13 Oct 2022 06:35:57 AM UTC

System load:  0.21          Processes:    235
Usage of /:   32.7% of 13.67GB Users logged in:  1
Memory usage: 28%          IPv4 address for ens33: 172.16.183.139
Swap usage:   0%

11 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.

Last login: Thu Oct 13 06:31:11 2022 from 172.16.183.1
visheshrangwani@visheshrangwani: ~$
```

Tring simultaneously from 2 terminals

```
— visheshrangwani@visheshrangwani: ~ — ssh visheshrangwani@172.16.183.139
Last login: Thu Oct 13 12:09:47 on ttys006
visheshrangwani@Visheshs-MacBook-Air ~ % knock -v 172.16.183.139 5010 6009 5012 7008
hitting tcp 172.16.183.139:5010
hitting tcp 172.16.183.139:6009
hitting tcp 172.16.183.139:5012
hitting tcp 172.16.183.139:7008
visheshrangwani@Visheshs-MacBook-Air ~ % ssh visheshrangwani@172.16.183.139
visheshrangwani@172.16.183.139's password:
Welcome to Ubuntu 20.04.5 LTS (GNU/Linux 5.4.0-128-generic x86_64)

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

System information as of Thu 13 Oct 2022 06:41:32 AM UTC

System load:  0.09          Processes:    237
Usage of /:   32.7% of 13.67GB Users logged in:  1
Memory usage: 28%          IPv4 address for ens33: 172.16.183.139
Swap usage:   0%

11 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.

Last login: Thu Oct 13 06:40:08 2022 from 172.16.183.1
visheshrangwani@visheshrangwani: ~$
```

```
— -zsh
Last login: Thu Oct 13 12:11:15 on ttys003
visheshrangwani@Visheshs-MacBook-Air ~ % ssh visheshrangwani@172.16.183.139
ssh: connect to host 172.16.183.139 port 22: Operation timed out
visheshrangwani@Visheshs-MacBook-Air ~ %
```

The second request is not fulfilled as the timeout for running stop command is very small and practically we can get just one SSH access.

Answer to part b:

One should prefer using TCP compared to UDP as:

- Reliable: TCP is more reliable without any packet losses. UDP does not ensure reliability in delivery of packets. If UDP packets are used, it may be possible that while knocking, one of the packets for a port in the sequence may get lost. In this case, we will not be able to knock the server. It might be that we try to find different errors which do not even exist without realizing that a possible reason for an unsuccessful knock could be a UDP packet loss.
- We know that the UDP packets are identified by just their destination port and destination address whereas the TCP packets also need source IP and port as well. There could be a scenario where a few UDP packets come in the same order as that of knocking within the sequence_timeout. In that case, the server won't be able to determine whether all these were from the same client or not and would render the connection open.
- Another advantage of TCP is that we can specify TCP flags such as ack, syn, fin, etc which must be set before considering that packet as a part of the knock sequence. This gives a higher degree of control in allowing which packets are to be considered for knocking and which not. This also ensures that random packets are not considered a part of the knock sequence.

We can specify the packet type (tcp/udp) while mentioning the sequence in knockd.conf file. For example, 2222:udp,3333:tcp,4444:udp; this requires that first packet must be UDP packet for port 222, second packet should be TCP to port 3333 and third should be UDP packet for port 444 for a successful knock.

Answer for part c:

The default ports in knockd.conf file are
7000,8000,9000 — for opening
9000,8000,7000 — for closing

Obviously, whenever one installs knockd and enables firewall, this configuration of port sequences is set by default. This is known to everyone. So any attacker would obviously first try this knock sequence to get access to a server. So it is highly unsafe as anyone can close and open the ports whenever they wish. Instead it's better to use a random port sequence (allowable upto a sequence of 7 ports).

References:

<https://linux.die.net/man/1/knockd>

<https://www.geeksforgeeks.org/iptables-command-in-linux-with-examples/>

<https://linux.die.net/man/8/iptables>

https://www.diffen.com/difference/TCP_vs_UDP