Port Knocking:

I have used a docker setup for port knocking.

- First, I created a new docker container by executing the below command:
  sudo docker run -dit -p 80:80 -p 443:443 -p 22:22 -p 7000:7000 -p 8000:8000 -p 9000:9000 --name port_knocking ubuntu

- Exec into the docker container:
  sudo docker exec -it --privileged port_knocking /bin/bash

- Installing the required packages:
  apt update && apt install iptables knockd net-tools sudo nano

- Setup OpenSSH server:
  service ssh start

- Edit the **[openSSH]** section in **/etc/knockd.conf:**
  [openSSH]
  sequence = 7000,8000,9000
  seq_timeout = 5
  command = /sbin/iptables -D INPUT -j DROP && /sbin/iptables -A INPUT -s %IP% -p tcp --dport 22 -j ACCEPT && /sbin/iptables -A INPUT -j DROP
  tcpflags = syn

- Adding the iptables rules:
  iptables -A INPUT -i lo -j ACCEPT
  iptables -A INPUT -i eth0 -p tcp -m tcp --dport 7000 -j ACCEPT
  iptables -A INPUT -i eth0 -p tcp -m tcp --dport 8000 -j ACCEPT
  iptables -A INPUT -i eth0 -p tcp -m tcp --dport 9000 -j ACCEPT
  iptables -A INPUT -j DROP

- Edited the **/etc/default/knockd** file and changes **START_KNOCKD=0 to START_KNOCKD=1** to enable the **knockd** service.

- Start the **knockd** service and add a user to perform ssh:
  service knockd start
  adduser {username}
  echo "{username} ALL=(ALL) NOPASSWD:ALL" >> /etc/sudoers
  Where username = akashk

```
root@db4dfc682323:/# adduser akashk
info: Adding user `akashk' ...
info: Selecting UID/GID from range 1000 to 59999 ...
info: Adding new group `akashk' (1001) ...
info: Adding new user `akashk' (1001) with group `akashk (1001)' ...
info: Creating home directory `/home/akashk' ...
info: Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
```

- Fetch the IP address of eth0 by executing the command:
  ifconfig eth0

```
root@db4dfc682323:/# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.17.0.2  netmask 255.255.0.0  broadcast 172.17.255.255
        ether 02:42:ac:11:00:02  txqueuelen 0  (Ethernet)
        RX packets 8555  bytes 65151757 (65.1 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 4905  bytes 344777 (344.7 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 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

  Noted IP address: 172.17.0.2

- Directly opening port 22 SSH is not working now:

```
akash@akash-Modern-14-B11MOU:~$ ssh akashk@172.17.0.2
ssh: connect to host 172.17.0.2 port 22: Connection refused
```

- Now by knocking 7000 8000 9000 sequence of ports, we are able to open and close port 22 SSH.

```
akash@akash-Modern-14-B11MOU:~$ ssh akashk@172.17.0.2
The authenticity of host '172.17.0.2 (172.17.0.2)' can't be established.
ED25519 key fingerprint is SHA256:91jwbTZ6G4oGvEWmHhFwBgt5sONIElyHerLJLlC8RO0.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '172.17.0.2' (ED25519) to the list of known hosts.
akashk@172.17.0.2's password:
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-52-generic x86_64)

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

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

akashk@db4dfc682323:~$ exit
logout
Connection to 172.17.0.2 closed.
```

- Lastly, we need to knock in an opposite sequence of ports to ensure no one can do ssh at port 22 again:

```
akash@akash-Modern-14-B11MOU:~$ knock -v 172.17.0.2 9000 8000 7000
hitting tcp 172.17.0.2:9000
hitting tcp 172.17.0.2:8000
hitting tcp 172.17.0.2:7000
akash@akash-Modern-14-B11MOU:~$ ssh akashk@172.17.0.2
^C
```

Part B: Why should one prefer doing this over TCP instead of UDP?

TCP is preferred for port knocking because:

1. **Reliable Delivery**: TCP ensures packets arrive in the correct order, unlike UDP, which may drop packets due to lack of connection state.
2. **Firewall Compatibility**: Many firewalls block or rate-limit UDP traffic, making it unreliable for port knocking.
3. **Ordered Delivery**: TCP ensures that port knocks are received in the correct sequence, while UDP packets can arrive out of order due to network conditions.

Part C: What is the default choice of ports in the knockd configuration? Is it safe?

The default ports in knockd.conf are 7000, 8000, and 9000.

Not entirely safe, as they are well-known defaults and may be detected by attackers using network monitoring tools and trying out all possible sequences.

To enhance security, use a custom port sequence and apply rate limits to mitigate brute-force knocking attempts.