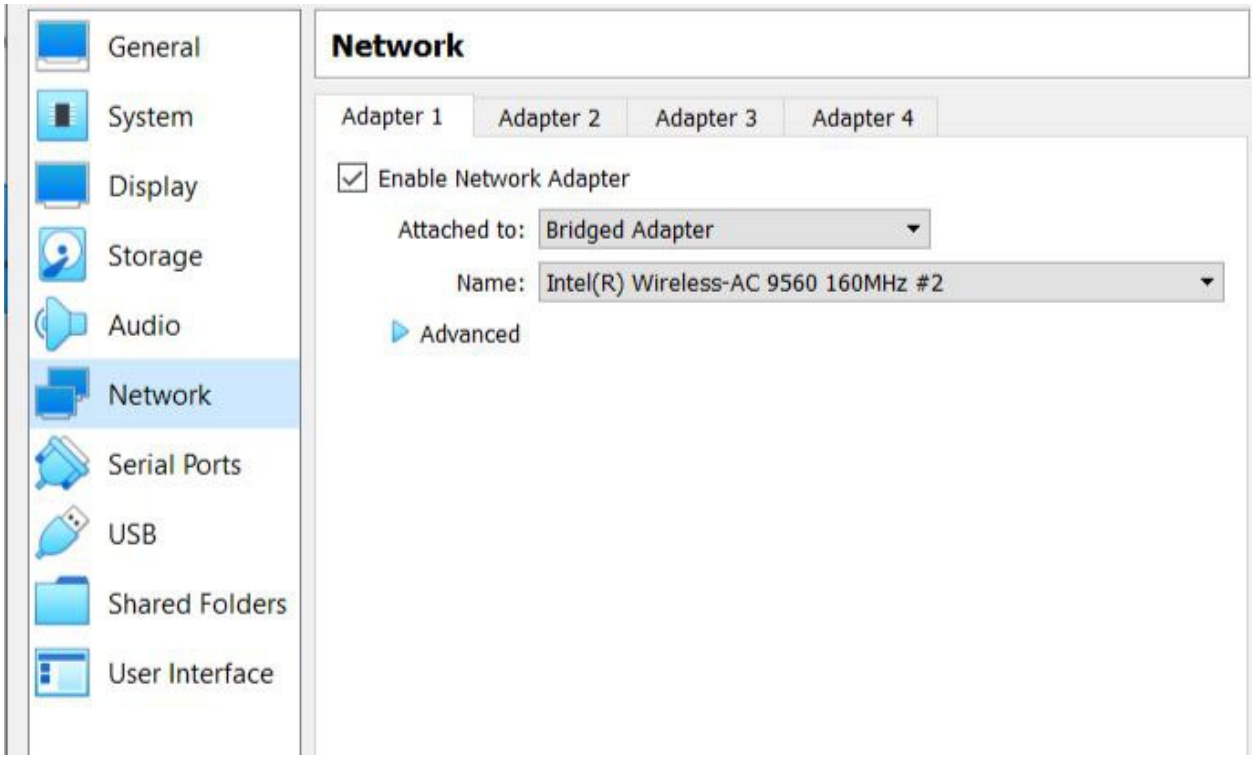
Lab 7: Iptables/Firewalling

Parth Kalkar

Q1. Set your interface on VM as bridged connection and restart interface (interface with network adapter BRIDGED ADAPTER). Install iptables with apt (if you don't have it). Check existing rules and display them.

- To do this task we can do the following steps :

    a. Installing iptables $ *sudo apt-get install iptables*
    b. List the current rules $ *sudo iptables -L*



- We get the following output

```
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
```

Chain FORWARD (policy ACCEPT)
target prot opt source destination
Chain OUTPUT (policy ACCEPT)
target prot opt source destination

```
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

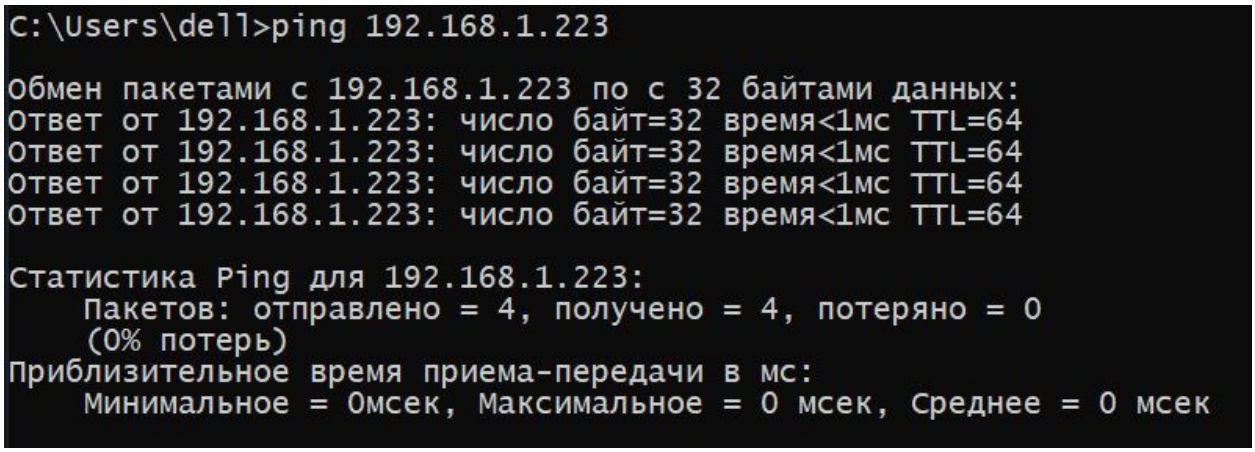
- Ping VM from host

```
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.223  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::7a25:494:20b1:7539  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:95:bd:d2  txqueuelen 1000  (Ethernet)
        RX packets 456  bytes 65163 (65.1 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 266  bytes 27511 (27.5 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.3.15  netmask 255.255.255.0  broadcast 10.0.3.255
        inet6 fe80::a185:9cb7:c32b:3f8  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:66:88:63  txqueuelen 1000  (Ethernet)
        RX packets 24  bytes 2254 (2.2 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 72  bytes 7665 (7.6 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 193  bytes 16380 (16.3 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 193  bytes 16380 (16.3 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

- Ping works correctly



Q2. Working with INPUT and OUTPUT chains:

- add a rule1 that blocks ping packets coming into the VM (your server with installed iptables).
- delete the rule1, and then add rule2 that blocks ping packets coming out of the VM - outgoing traffic, you can block specific range of ip (for example you can block specific IP range. The range that used your host machine).
- To test each of the above rules, try to ping the VM and observe whether it is successful or not.

Record your rules in the report, its short explanation and results of the implementations (acceptance testings)

1. Adding rule1 that blocks ping packets coming into the VM

   $ *sudo iptables -A INPUT -p icmp --icmp-type 8 -j DROP*

```
C:\Users\dell>ping 192.168.1.223

Обмен пакетами с 192.168.1.223 по с 32 байтами данных:
Превышен интервал ожидания для запроса.
Превышен интервал ожидания для запроса.
Превышен интервал ожидания для запроса.
Превышен интервал ожидания для запроса.

Статистика Ping для 192.168.1.223:
    Пакетов: отправлено = 4, получено = 0, потеряно = 4
    (100% потерь)
```

2. Deleting the rule1, and then add rule2 that blocks ping packets coming out of the VM

   $ *sudo iptables -D INPUT 1*

```
Chain INPUT (policy ACCEPT)
num  target     prot opt source               destination
1    DROP       icmp --  anywhere             anywhere             icmp echo-re
quest

Chain FORWARD (policy ACCEPT)
num  target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
num  target     prot opt source               destination
```

   $ *sudo iptables -A OUTPUT -p icmp --icmp-type echo-reply -d 192.168.1.42 -j DROP*

```
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

3. It does not reply to the IP of the host machine. If we change the blocked IP in the firewall then VM should reply to the host machine.

```
C:\Users\dell>ping 192.168.1.223

Обмен пакетами с 192.168.1.223 по с 32 байтами данных:
Превышен интервал ожидания для запроса.
Превышен интервал ожидания для запроса.
Превышен интервал ожидания для запроса.
Превышен интервал ожидания для запроса.

Статистика Ping для 192.168.1.223:
    Пакетов: отправлено = 4, получено = 0, потеряно = 4
    (100% потерь)
```

$ *sudo iptables -A OUTPUT -p icmp --icmp-type echo-reply -d 192.168.1.41 -j DROP*

```
C:\Users\dell>ping 192.168.1.223

Обмен пакетами с 192.168.1.223 по с 32 байтами данных:
Ответ от 192.168.1.223: число байт=32 время<1мс TTL=64
Ответ от 192.168.1.223: число байт=32 время<1мс TTL=64
Ответ от 192.168.1.223: число байт=32 время<1мс TTL=64
Ответ от 192.168.1.223: число байт=32 время<1мс TTL=64

Статистика Ping для 192.168.1.223:
    Пакетов: отправлено = 4, получено = 4, потеряно = 0
    (0% потерь)
Приблизительное время приема-передачи в мс:
    Минимальное = 0мсек, Максимальное = 0 мсек, Среднее = 0 мсек
```

4. Testing each of the above rules and trying to ping the VM. It will be blocked, but since the command of adding the rule doesn't include showing an error message, then it will not show the error message, yet the request would be blocked.

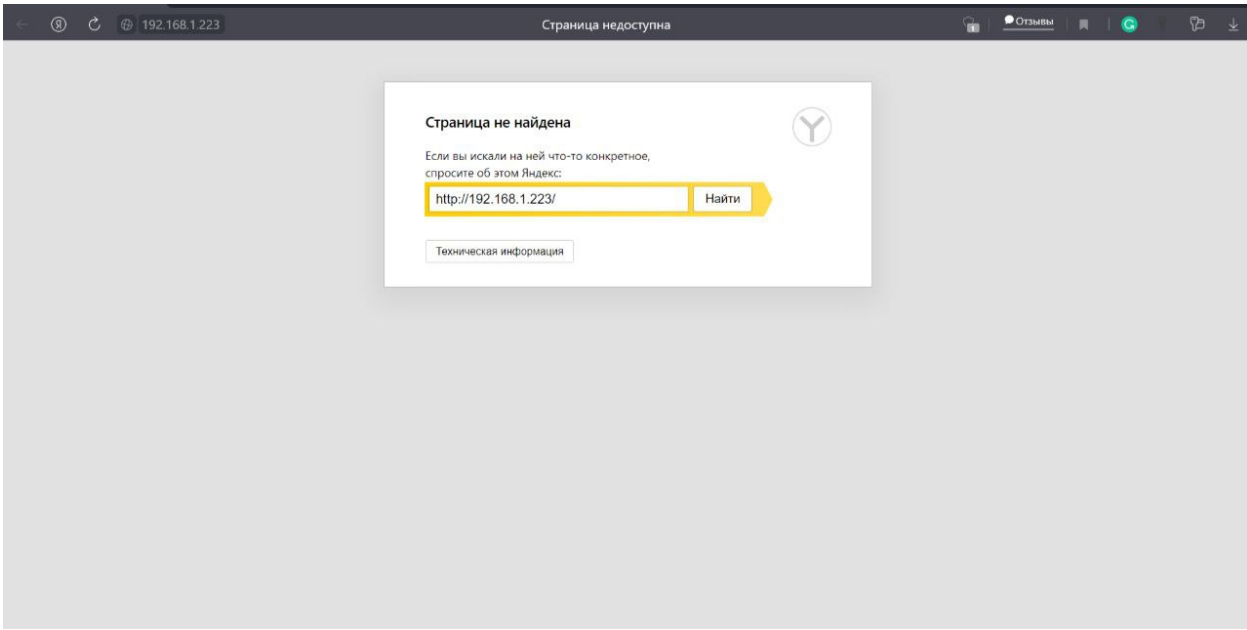
Q3. Install a web server on your VM (if you don't have it). Add a rule to the firewall that prevents loading your web server's IP from the host machine in the browser. Test it by trying to load a web page from the VM in your host machine's web browser (http://IP_address_of_your_vm).

- The solution is as follows

a. Load a page from the host machine in the browser. As a result we can access a web page from the host machine.

b. Add a rule to the firewall that prevents loading your web server's IP from the host machine in the browser.

To do this, $ *sudo iptables -A INPUT -p tcp --dport 80 -s 192.168.1.42 -j DROP*



c. The page of the VM Web Server cannot be accessed from the host machine but from the VM itself it can.

Q4. Create firewall rules that open ports for SMTP, DNS, POP3 and SSH connections in your VM and block others ports (for you to be able to work with mentioned services).
Hint: to test some of the services you can use netcat utility

Record your rules in the report, its short explanation and results of the implementations (acceptance testings)

- The solution is as follows:
  Steps -
  1. Root access
  # Set default chain policies
  $ *iptables -P INPUT DROP*
  $ *iptables -P FORWARD DROP*
  $ *iptables -P OUTPUT DROP*

  # Accept on localhost
  $ *iptables -A INPUT -i lo -j ACCEPT*

$ *iptables -A OUTPUT -o lo -j ACCEPT*

2. SMTP: port 25 - To allow your server to respond to SMTP connections

$ *sudo iptables -A INPUT -p tcp --dport 25 -m conntrack --ctstate NEW,ESTABLISHED -j ACCEPT*

$ *sudo iptables -A OUTPUT -p tcp --sport 25 -m conntrack --ctstate ESTABLISHED -j ACCEPT*

3. POP3: port 110 - To allow your server to respond to POP3 connections

$ *sudo iptables -A INPUT -p tcp --dport 110 -m conntrack --ctstate NEW,ESTABLISHED -j ACCEPT*

$ *sudo iptables -A OUTPUT -p tcp --sport 110 -m conntrack --ctstate ESTABLISHED -j ACCEPT*

4. SSH: port 22 - To allow your server to respond to POP3 connections

$ *sudo iptables -A OUTPUT -p tcp --dport 22 -m conntrack --ctstate NEW,ESTABLISHED -j ACCEPT*

$ *sudo iptables -A INPUT -p tcp --sport 22 -m conntrack --ctstate ESTABLISHED -j ACCEPT*

5. DNS: port 53 - To allow your server to respond to DNS connections

$ *sudo iptables -A OUTPUT -p udp --dport 53 -m conntrack --ctstate ESTABLISHED -j ACCEPT*

$ *sudo iptables -A INPUT -p udp --dport 53 -m conntrack --ctstate ESTABLISHED -j ACCEPT*

a. To check that the commands are working fine, let's install open-ssh on the VM machine and try to access port 22 from the host machine.

   To install openssh, use the command: $ *sudo apt install openssh-server*

   After installation the ssh service will run on port 22. Now let's try to access this service from the host machine using *telnet*.

```
C:\Users\dell>telnet 192.168.1.223 22
```

b. Also we can install pop3 on the VM machine and try to access port 110 from the host machine.

   To install pop3, use the command $ *sudo apt install dovecot-pop3d*. After installation POP3 service is running on 110 ports. Let's try to access it from the host machine.

```
C:\Users\dell>telnet 192.168.1.223 110
```

c. Finally, let's try to access 80 ports to check if our firewall works correctly.

```
C:\Users\dell>telnet 192.168.1.223 80
Подключение к 192.168.1.223...Не удалось открыть подключение к этому узлу, на порт 80: Сбой подключения

C:\Users\dell>
```

It is not accessible. Hence, our firewall works correctly!