LAB06 Linux Firewall exercise

Class	M02
Student ID	B2014926
Name	Tran Dang Khoa
Email address	Khoab2014926@student.ctu.edu.vn
Class	
Browser	Safari, Chrome, IE, Firefox

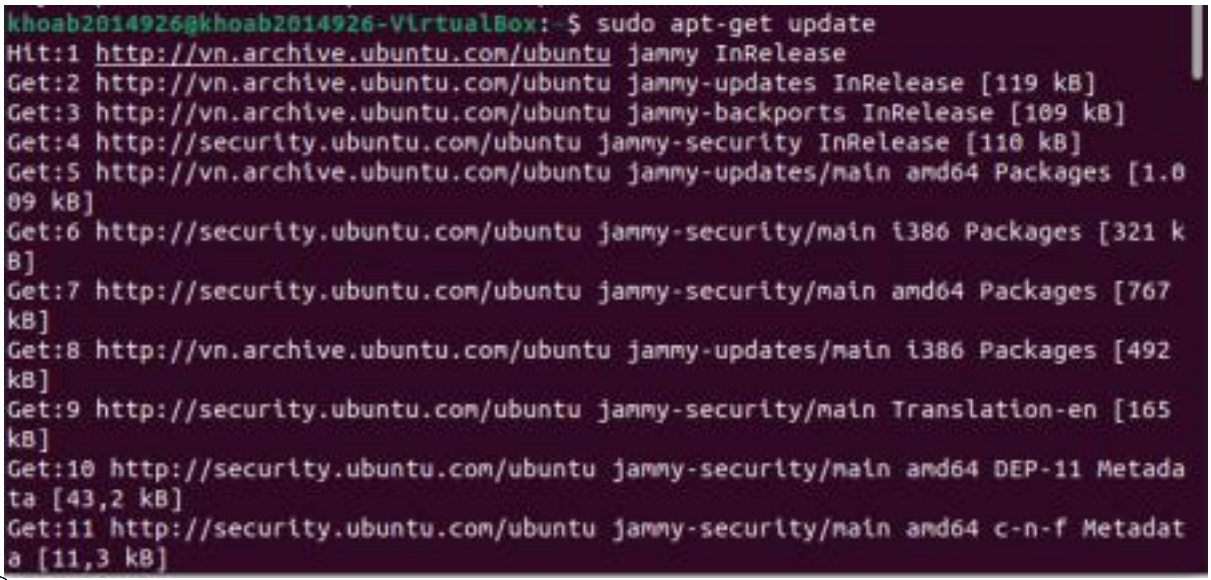
Exercise following command, explain the command

Step 1 - Installing Iptables

① Update the package list by running the following command:

Explain: ‘sudo apt-get update’ is used to update system's knowledge of available software packages, which is essential before performing package installations or upgrades to ensure we are working with the latest package information

$sudo apt-get update

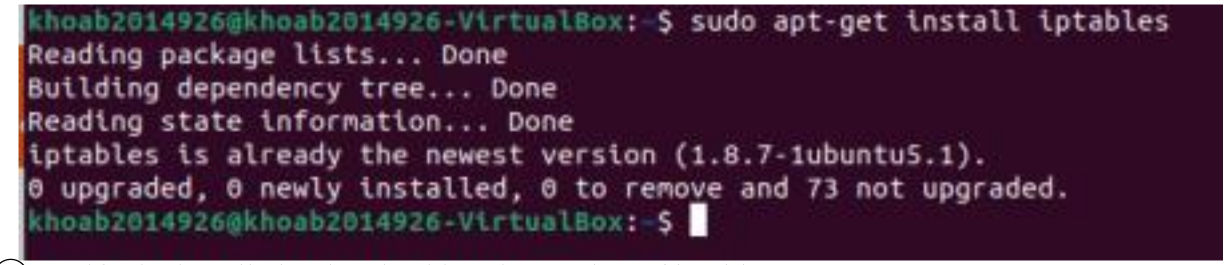


```
khoab2014926@khoab2014926-VirtualBox:~$ sudo apt-get update
Hit:1 http://vn.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://vn.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:3 http://vn.archive.ubuntu.com/ubuntu jammy-backports InRelease [109 kB]
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:5 http://vn.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [1.0
09 kB]
Get:6 http://security.ubuntu.com/ubuntu jammy-security/main i386 Packages [321 k
B]
Get:7 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [767
kB]
Get:8 http://vn.archive.ubuntu.com/ubuntu jammy-updates/main i386 Packages [492
kB]
Get:9 http://security.ubuntu.com/ubuntu jammy-security/main Translation-en [165
kB]
Get:10 http://security.ubuntu.com/ubuntu jammy-security/main amd64 DEP-11 Metada
ta [43,2 kB]
Get:11 http://security.ubuntu.com/ubuntu jammy-security/main amd64 c-n-f Metadat
a [11,3 kB]
```

② Install iptables by running the following command:

Explain: ‘sudo apt-get install iptables’ installs the "iptables" package, which allows to configure and manage the firewall rules on your Linux system

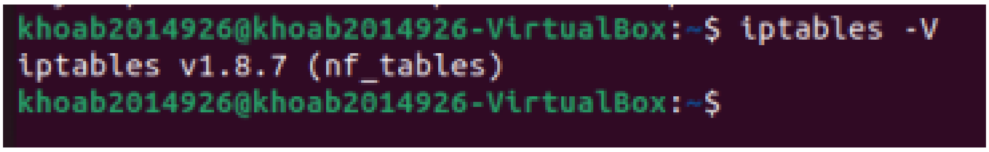
$sudo apt-get install iptables



```
khoab2014926@khoab2014926-VirtualBox:~$ sudo apt-get install iptables
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
iptables is already the newest version (1.8.7-1ubuntu5.1).
0 upgraded, 0 newly installed, 0 to remove and 73 not upgraded.
khoab2014926@khoab2014926-VirtualBox:~$
```

③ Verify the installation by checking the version of iptables:

$iptables –version or $iptables -V (example below)



```
khoab2014926@khoab2014926-VirtualBox:~$ iptables -V
iptables v1.8.7 (nf_tables)
khoab2014926@khoab2014926-VirtualBox:~$
```

④ Show a list of all the rules in the fire wall

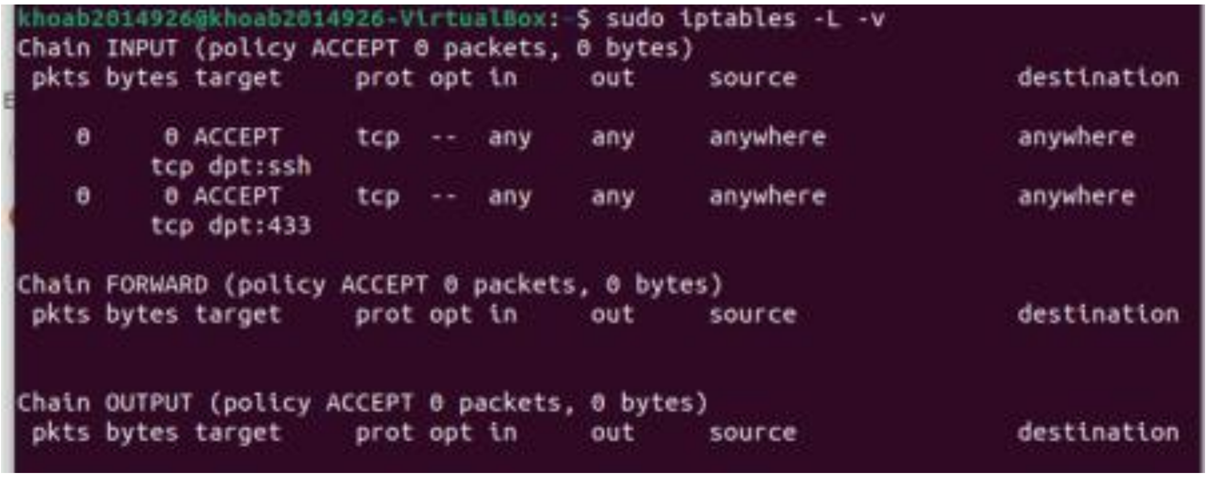
$sudo iptables –L –v

Explain:

iptables: command to manage the iptables firewall

-L or --list: This option is used to list and display the firewall rules for each chain (INPUT, OUTPUT, FORWARD, etc.).

-v or --verbose: This option provides more detailed information about the rules, including packet and byte counters.



```
khoab2014926@khoab2014926-VirtualBox:~$ sudo iptables -L -v
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

    0     0 ACCEPT     tcp  --  any    any     anywhere             anywhere
         tcp dpt:ssh
    0     0 ACCEPT     tcp  --  any    any     anywhere             anywhere
         tcp dpt:433

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination


Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
```

Step 2 - Defining Chain Rules

⑤ Insert the –A option (Append) right after the iptables command:

$sudo iptables -A

Explain:
With some option:

+ -i (interface) – the network interface you wish to filter traffic from

+ -p (protocol) – the network protocol where your

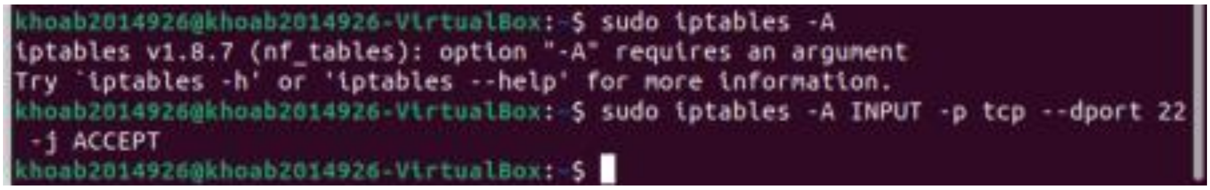
filtering process takes place

+ -s (source) – the address from which traffic comes from. May be a

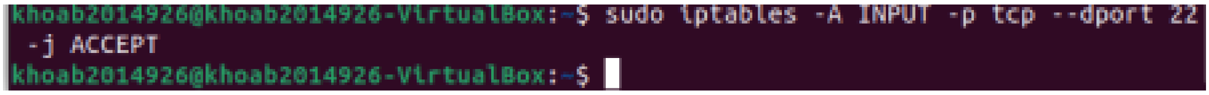
hostname or IP address

+ --dport (destination port) – the destination port number of a protocol, example 22 (SSH), 433 (https),...

+ -j (target) – the target name (ACCEPT, DROP, RETURN). You should to insert every time you make new rule



```
khoab2014926@khoab2014926-VirtualBox:~$ sudo iptables -A
iptables v1.8.7 (nf_tables): option "-A" requires an argument
Try `iptables -h' or 'iptables --help' for more information.
khoab2014926@khoab2014926-VirtualBox:~$ sudo iptables -A INPUT -p tcp --dport 22
 -j ACCEPT
khoab2014926@khoab2014926-VirtualBox:~$ 
```



```
khoab2014926@khoab2014926-VirtualBox:~$ sudo iptables -A INPUT -p tcp --dport 22
 -j ACCEPT
khoab2014926@khoab2014926-VirtualBox:~$ 
```

Enabling traffic on Localhost

⑥ The `-i lo` option specifies the loopback interface, and the `-j ACCEPT` option allows the traffic

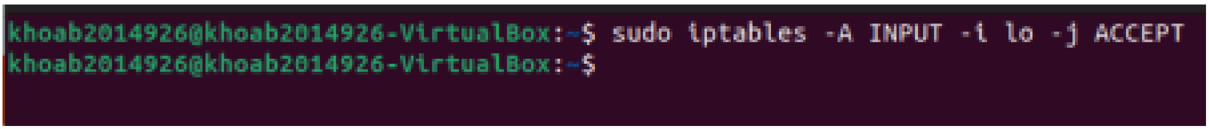
$sudo iptables –A INPUT –i lo –j ACCEPT

Explain:

-A INPUT: This specifies that you want to append (add) a rule to the INPUT chain.

-i lo: It specifies the network interface to which the rule applies, in this case, the loopback interface (lo).

-j ACCEPT: This part of the rule specifies the action to take when traffic matches the rule, which is to ACCEPT the traffic.



```
khoab2014926@khoab2014926-VirtualBox:~$ sudo iptables -A INPUT -i lo -j ACCEPT
khoab2014926@khoab2014926-VirtualBox:~$
```

⑦ Check that the new rule was added:

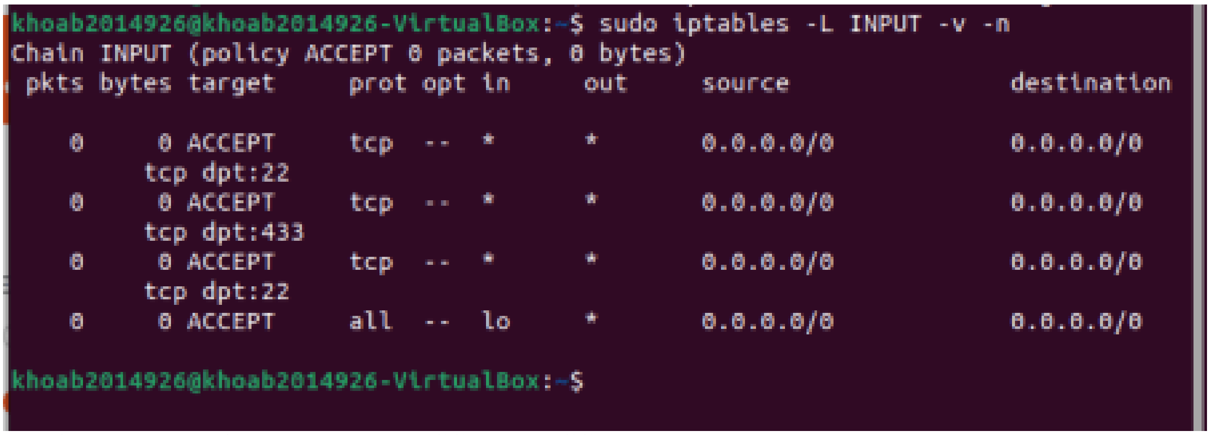
$sudo iptables –L INPUT –v –n

Explain:

-L INPUT: This specifies that you want to list and display the rules for the INPUT chain.

-v or --verbose: This option provides more detailed information about the rules, including packet and byte counters.

-n or --numeric: This option displays IP addresses and port numbers in numeric format instead of resolving them to hostnames and service names.



```
khoab2014926@khoab2014926-VirtualBox:~$ sudo iptables -L INPUT -v -n
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0
         tcp dpt:22
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0
         tcp dpt:433
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0
         tcp dpt:22
    0     0 ACCEPT     all  --  lo     *       0.0.0.0/0            0.0.0.0/0

khoab2014926@khoab2014926-VirtualBox:~$
```

Enabling conntections on HTTP, SSH and SSL port

⑧ To enable incoming connecton on HTTP (port 80), SSH (port 22), and SSL (port 443) using iptables on Ubuntu. Add rules to allow incoming traffic on HTTP, SSH, and SSL port:

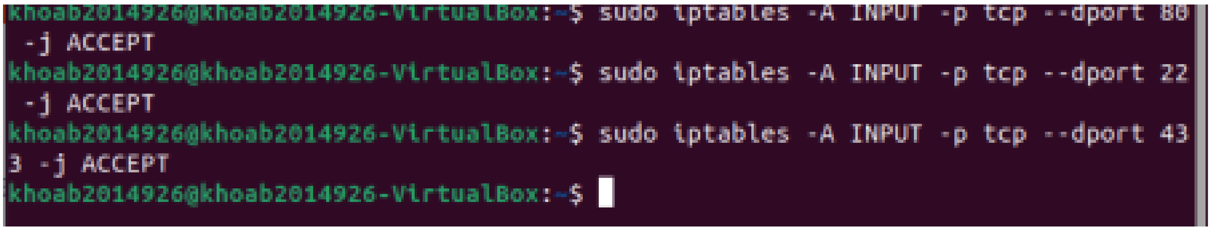
$sudo iptables –A INPUT –p tcp –dport 80 –j ACCEPT

$sudo iptables –A INPUT –p tcp –dport 22 –j ACCEPT

$sudo iptables –A INPUT –p tcp –dport 443 –j ACCEPT

Explain:

These commands allow incoming TCP traffic on port 80, 22 and 443, which is commonly used for web server traffic. It's a common rule in firewall configurations to allow HTTP traffic to reach a web server hosted on the system



```
khoab2014926@khoab2014926-VirtualBox:~$ sudo iptables -A INPUT -p tcp --dport 80
 -j ACCEPT
khoab2014926@khoab2014926-VirtualBox:~$ sudo iptables -A INPUT -p tcp --dport 22
 -j ACCEPT
khoab2014926@khoab2014926-VirtualBox:~$ sudo iptables -A INPUT -p tcp --dport 43
3 -j ACCEPT
khoab2014926@khoab2014926-VirtualBox:~$ 
```

⑨ And use command to check:

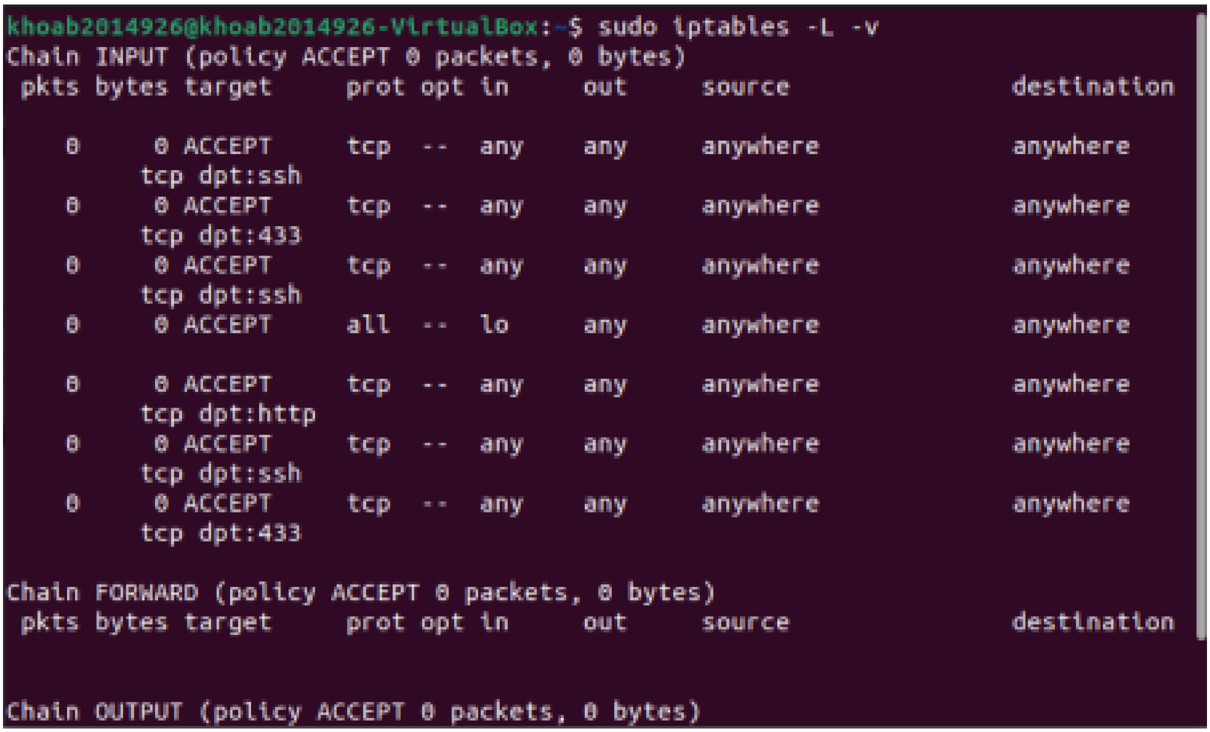
$sudo iptables –L –v

Explain:

-L or --list: This option specifies that you want to list and display the rules for all chains (INPUT, OUTPUT, FORWARD, etc.).

-v or --verbose: This option provides more detailed information about the rules, including packet and byte counters.

Get a detailed listing of the current firewall rules for all chains, including rule numbers, target actions, protocol, source and destination IP addresses, source and destination ports, and packet and byte counters. This command is useful for inspecting the current firewall configuration and monitoring traffic statistics



```
khoab2014926@khoab2014926-VirtualBox:~$ sudo iptables -L -v
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

    0     0 ACCEPT     tcp  --  any    any     anywhere             anywhere
         tcp dpt:ssh
    0     0 ACCEPT     tcp  --  any    any     anywhere             anywhere
         tcp dpt:433
    0     0 ACCEPT     tcp  --  any    any     anywhere             anywhere
         tcp dpt:ssh
    0     0 ACCEPT     all  --  lo     any     anywhere             anywhere

    0     0 ACCEPT     tcp  --  any    any     anywhere             anywhere
         tcp dpt:http
    0     0 ACCEPT     tcp  --  any    any     anywhere             anywhere
         tcp dpt:ssh
    0     0 ACCEPT     tcp  --  any    any     anywhere             anywhere
         tcp dpt:433

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination


Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
```

Filtering packets based on Source

⑩ Filter packets based on source IP addresses in iptables by adding a rule that matches packets based on their source address.

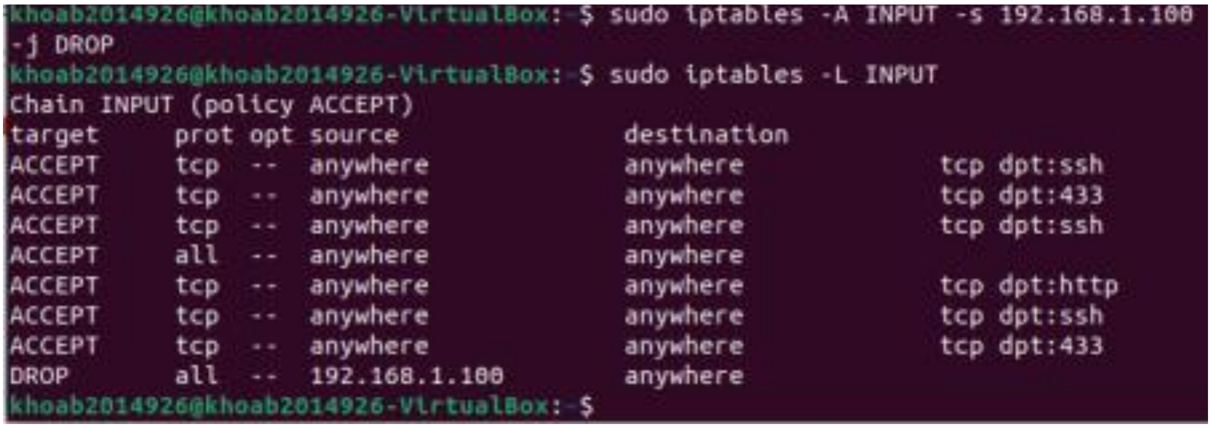
$sudo iptables –A INPUT –s 192.168.1.100 –j DROP

Explain:

--s 192.168.1.100: This specifies the source IP address for the rule, which is 192.168.1.100. Any incoming traffic originating from this IP address will be affected by this rule.

-j DROP: This part of the rule specifies the action to take when traffic matches the rule, which is to DROP (block) the traffic.

This command blocks all incoming traffic from the IP address 192.168.1.100 by appending a rule to the INPUT chain that drops any packets coming from that specific source IP address. This can be used to restrict or deny incoming connections from a particular source.



```
khoab2014926@khoab2014926-VirtualBox:~$ sudo iptables -A INPUT -s 192.168.1.100
-j DROP
khoab2014926@khoab2014926-VirtualBox:~$ sudo iptables -L INPUT
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:ssh
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:433
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:ssh
ACCEPT     all  --  anywhere             anywhere
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:http
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:ssh
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:433
DROP       all  --  192.168.1.100        anywhere
khoab2014926@khoab2014926-VirtualBox:~$
```

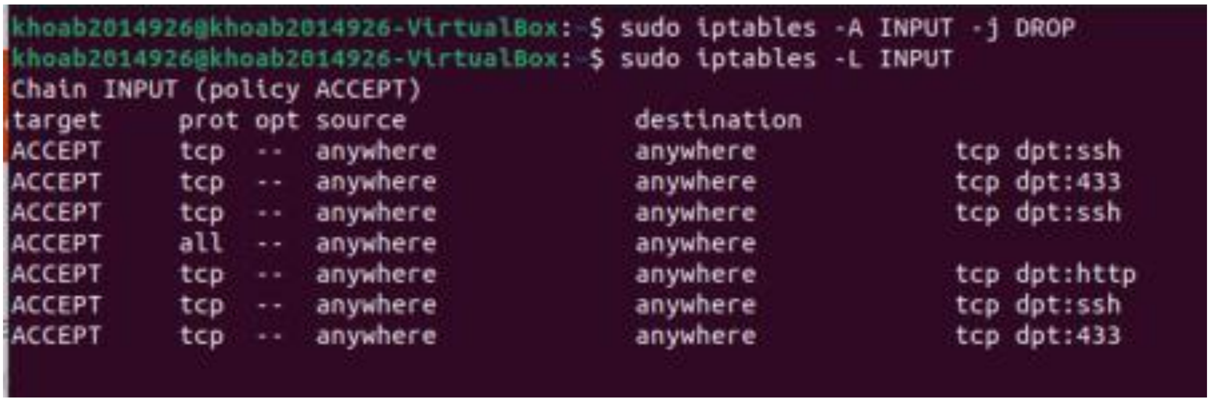
Dropping all Other traffic

⑪ Add a default DROP rule to the firewall's INPUT, OUTPUT, and FORWARD chains to drop all other traffic.

$sudo iptables –A INPUT –j DROP

Explain:

This command effectively blocks all incoming traffic to your system by appending a rule to the INPUT chain that drops all packets

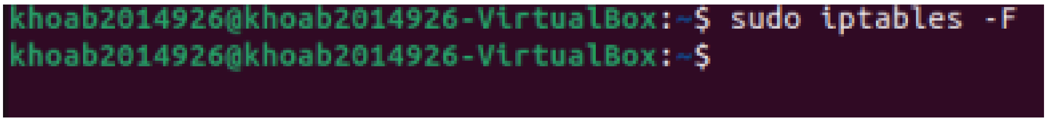


```
khoab2014926@khoab2014926-VirtualBox:~$ sudo iptables -A INPUT -j DROP
khoab2014926@khoab2014926-VirtualBox:~$ sudo iptables -L INPUT
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:ssh
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:433
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:ssh
ACCEPT     all  --  anywhere             anywhere
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:http
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:ssh
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:433
```

Deleting rules

⑫ Remove all rules and start with a clean slate, you can use the option –F (flush):

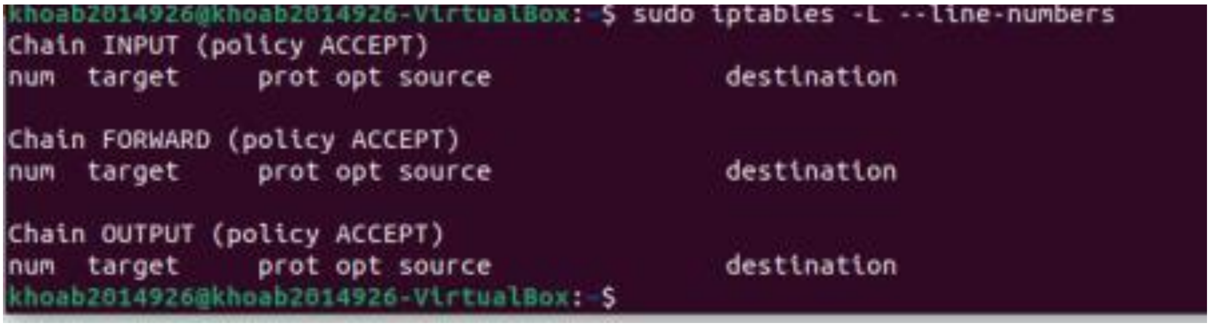
$sudo iptables –F



```
khoab2014926@khoab2014926-VirtualBox:~$ sudo iptables -F
khoab2014926@khoab2014926-VirtualBox:~$
```

⑬ To delete a specfc rule in iptables, use the `iptables –D` command

$sudo iptables –L –line-numbers

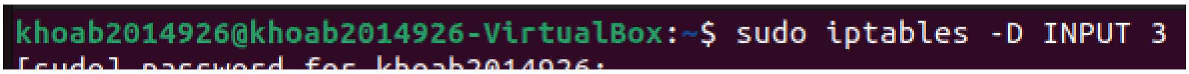


```
khoab2014926@khoab2014926-VirtualBox:~$ sudo iptables -L --line-numbers
Chain INPUT (policy ACCEPT)
num  target     prot opt source               destination

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

Chain OUTPUT (policy ACCEPT)
num  target     prot opt source               destination
khoab2014926@khoab2014926-VirtualBox:~$
```

⑭ Delete the rule with command

$sudo iptables –D <chain> <line_number>

$sudo iptables –D INPUT 3



```
khoab2014926@khoab2014926-VirtualBox:~$ sudo iptables -D INPUT 3
[sudo] password for khoab2014926:
```

Step3 - Persisting Changes

⑮ To make these **changes presistent** after restarting the server:

$sudo /sbin/iptables-save

Explain: This command is used to save the current iptables rules

⑯ Disable iptables, we need to execute following commands:

$sudo iptable –F

$sudo /sbin/iptables-save