

Assignment 5

G.Rama Rohit Reddy - 201525083

Everything as root

1. Create rules to allow only ssh and port 80 traffic:

Sol:

- 1) iptables -A INPUT -p tcp -m multiport --dports 22,80 -j ACCEPT
- 2) iptables -A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
- 3) iptables -A INPUT -j REJECT --reject-with icmp-port-unreachable

```
root@osboxes:/home/osboxes# iptables -n -L
Chain INPUT (policy ACCEPT)
target     prot opt source                destination
ACCEPT     icmp -- 0.0.0.0/0              0.0.0.0/0
ACCEPT     tcp  -- 0.0.0.0/0              0.0.0.0/0          multiport dports 2
2,80
ACCEPT     all  -- 0.0.0.0/0              0.0.0.0/0          state RELATED,ESTA
BLISHED
REJECT     all  -- 0.0.0.0/0              0.0.0.0/0          reject-with icmp-p
ort-unreachable
Chain FORWARD (policy ACCEPT)
target     prot opt source                destination
Chain OUTPUT (policy ACCEPT)
target     prot opt source                destination
root@osboxes:/home/osboxes#
```

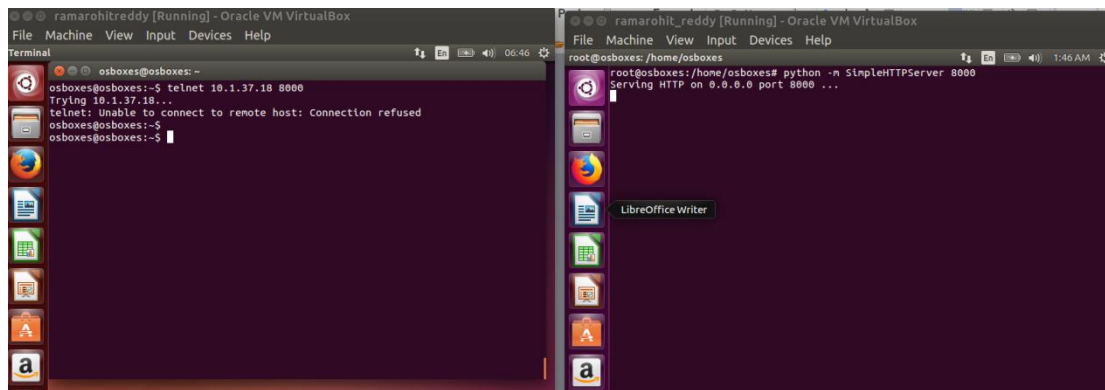
Effect of adding rules:

Only ssh(port 22) and port 80 can be access other port are not.

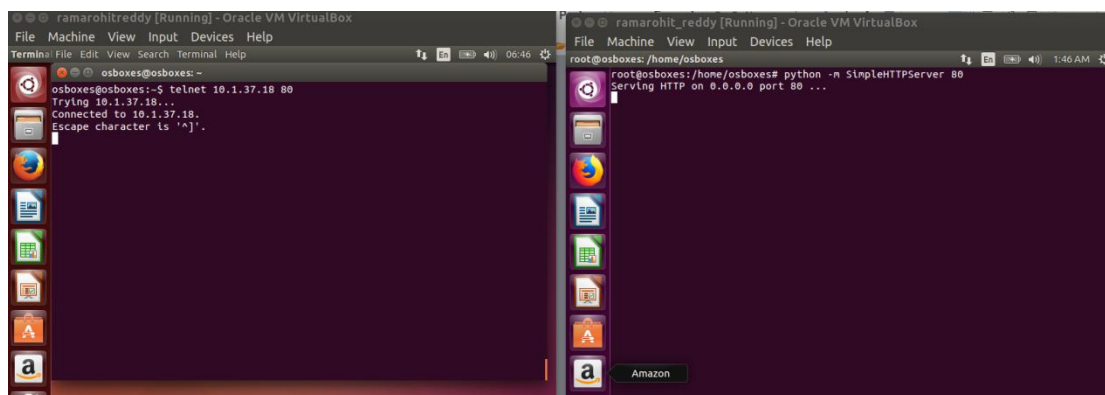
```
osboxes@osboxes:~$ ssh 10.1.37.18
The authenticity of host '10.1.37.18 (10.1.37.18)' can't be established.
ECDSA key fingerprint is 35:b5:4f:d2:07:15:be:ff:f0:b2:2c:5f:bb:98:a9:08.
Are you sure you want to continue connecting (yes/no)?
```

We can see client can access ssh.

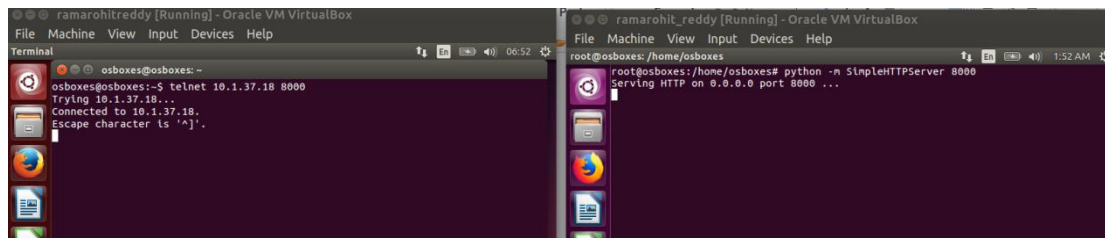
But it cannot access port 8000 as it we allowed only ssh and port 80 traffic.



It allows port 80 traffic/access.



Initially without these iptable rules it allows access to all ports.



2) Create rules to accept traffic only on a range of ports.

Sol:

- 1) iptables -A INPUT -p tcp -m multiport --dports 80:143 -j ACCEPT
- 2) iptables -A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
- 3) iptables -A INPUT -j REJECT --reject-with icmp-port-unreachable

```

root@osboxes:/home/osboxes# iptables -n -L
Chain INPUT (policy ACCEPT)
target     prot opt source                destination          multiport dports 8
0:143
ACCEPT     tcp  --  0.0.0.0/0              0.0.0.0/0            multiport dports 8
0:143
ACCEPT     all  --  0.0.0.0/0              0.0.0.0/0            state RELATED,ESTA
BLISHED
REJECT     all  --  0.0.0.0/0              0.0.0.0/0            reject-with icmp-p
ort-unreachable

Chain FORWARD (policy ACCEPT)
target     prot opt source                destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source                destination
root@osboxes:/home/osboxes#

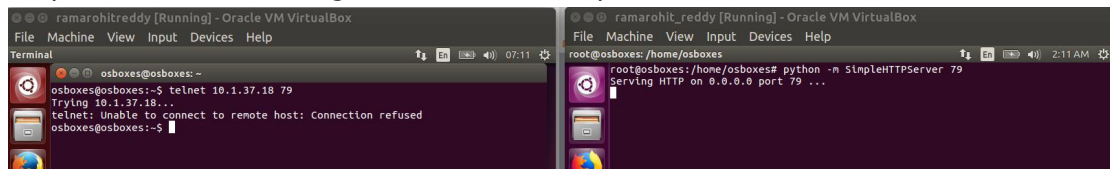
```

This only accepts connections/traffic in the range of 80-143 ports.

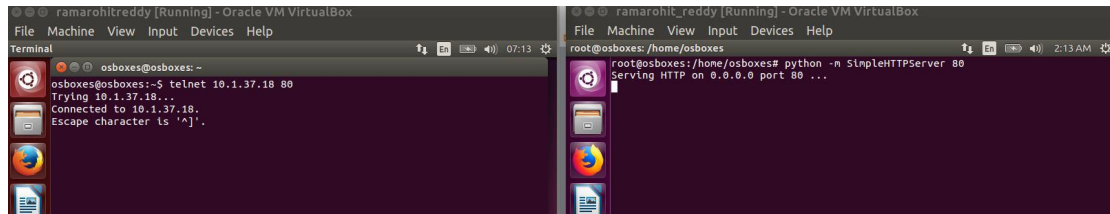
Effect of adding rules:

It does not accept connections out of range [80,143]

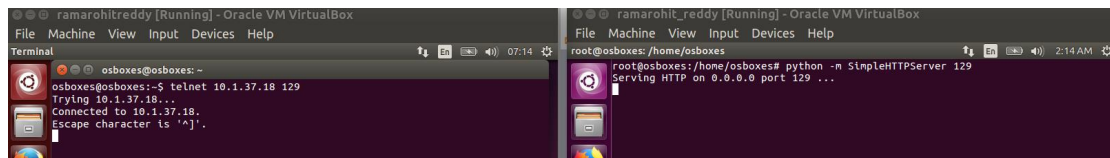
As port 79 is out of range it does not accept.



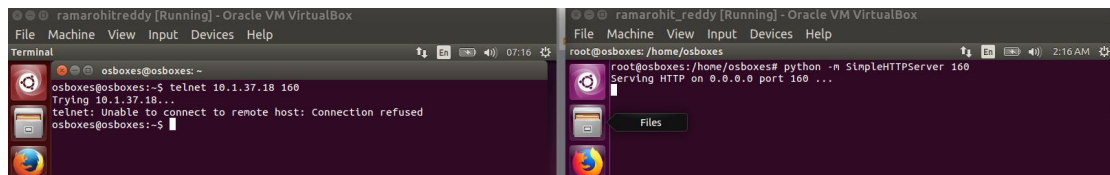
As port 80 is in range it accepts



As port 129 is in range it accepts



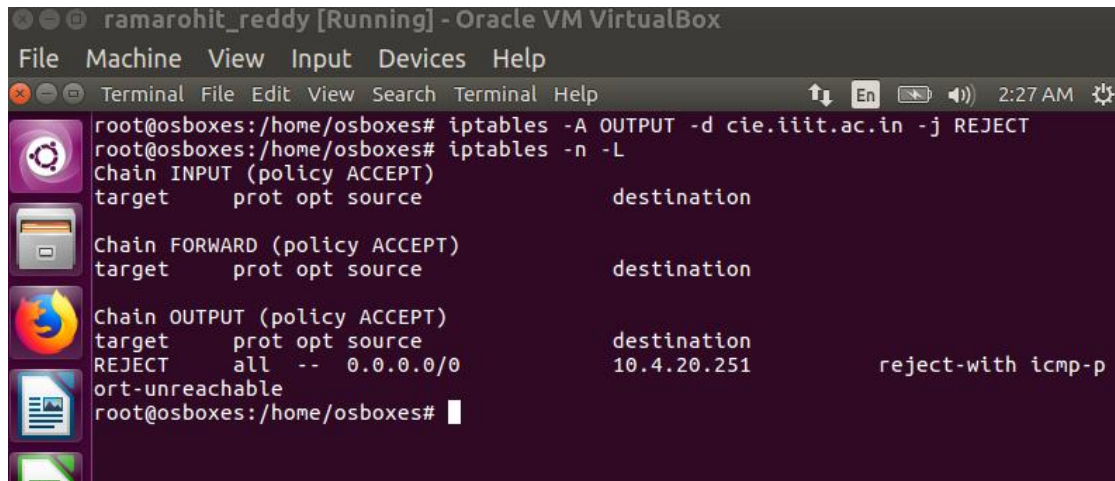
As port 160 is out of range it does not accept



3) Add rules to deny packets to a specific domain (like cie.iiit.ac.in)

Sol:

1) iptables -A OUTPUT -d cie.iiit.ac.in -j REJECT

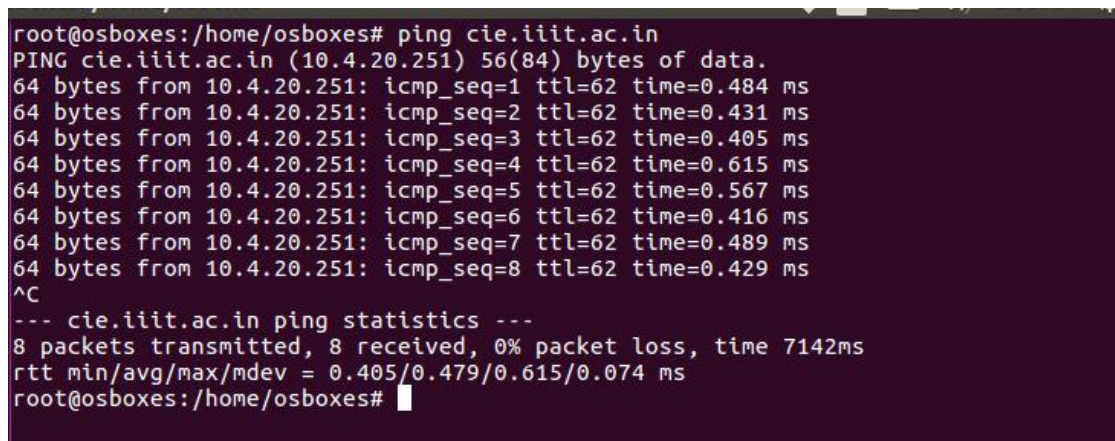


```
ramarohit_reddy [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminal File Edit View Search Terminal Help 2:27 AM
root@osboxes:/home/osboxes# iptables -A OUTPUT -d cie.iiit.ac.in -j REJECT
root@osboxes:/home/osboxes# iptables -n -L
Chain INPUT (policy ACCEPT)
target    prot opt source                destination
Chain FORWARD (policy ACCEPT)
target    prot opt source                destination
Chain OUTPUT (policy ACCEPT)
target    prot opt source                destination
REJECT    all  --  0.0.0.0/0             10.4.20.251          reject-with icmp-p
ort-unreachable
root@osboxes:/home/osboxes#
```

It denies packets to cie.iiit.ac.in.

Effect of Adding:

Before:



```
root@osboxes:/home/osboxes# ping cie.iiit.ac.in
PING cie.iiit.ac.in (10.4.20.251) 56(84) bytes of data.
64 bytes from 10.4.20.251: icmp_seq=1 ttl=62 time=0.484 ms
64 bytes from 10.4.20.251: icmp_seq=2 ttl=62 time=0.431 ms
64 bytes from 10.4.20.251: icmp_seq=3 ttl=62 time=0.405 ms
64 bytes from 10.4.20.251: icmp_seq=4 ttl=62 time=0.615 ms
64 bytes from 10.4.20.251: icmp_seq=5 ttl=62 time=0.567 ms
64 bytes from 10.4.20.251: icmp_seq=6 ttl=62 time=0.416 ms
64 bytes from 10.4.20.251: icmp_seq=7 ttl=62 time=0.489 ms
64 bytes from 10.4.20.251: icmp_seq=8 ttl=62 time=0.429 ms
^C
--- cie.iiit.ac.in ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7142ms
rtt min/avg/max/mdev = 0.405/0.479/0.615/0.074 ms
root@osboxes:/home/osboxes#
```


It denies packets from cie.iiit.ac.in

```
From 10.1.37.18 icmp_seq=5 Destination Port Unreachable
From 10.1.37.18 icmp_seq=5 Destination Port Unreachable
From 10.1.37.18 icmp_seq=5 Destination Port Unreachable
From 10.1.37.18 icmp_seq=5 Destination Port Unreachable
From 10.1.37.18 icmp_seq=5 Destination Port Unreachable
From 10.1.37.18 icmp_seq=5 Destination Port Unreachable
From 10.1.37.18 icmp_seq=5 Destination Port Unreachable
From 10.1.37.18 icmp_seq=5 Destination Port Unreachable
From 10.1.37.18 icmp_seq=5 Destination Port Unreachable
From 10.1.37.18 icmp_seq=5 Destination Port Unreachable
From 10.1.37.18 icmp_seq=5 Destination Port Unreachable
From 10.1.37.18 icmp_seq=5 Destination Port Unreachable
From 10.1.37.18 icmp_seq=5 Destination Port Unreachable
From 10.1.37.18 icmp_seq=5 Destination Port Unreachable
From 10.1.37.18 icmp_seq=5 Destination Port Unreachable
From 10.1.37.18 icmp_seq=5 Destination Port Unreachable
From 10.1.37.18 icmp_seq=5 Destination Port Unreachable
From 10.1.37.18 icmp_seq=5 Destination Port Unreachable
From 10.1.37.18 icmp_seq=5 Destination Port Unreachable
From 10.1.37.18 icmp_seq=5 Destination Port Unreachable
LibreOffice Impress mp_seq=5 Destination Port Unreachable
From 10.1.37.18 icmp_seq=5 Destination Port Unreachable
From 10.1.37.18 icmp_seq=5 Destination Port Unreachable
From 10.1.37.18 icmp_seq=5 Destination Port Unreachable
From 10.1.37.18 icmp_seq=5 Destination Port Unreachable
From 10.1.37.18 icmp_seq=5 Destination Port Unreachable
From 10.1.37.18 icmp_seq=5 Destination Port Unreachable
ping: sendmsg: Operation not permitted
^C
--- cie.iit.ac.in ping statistics ---
5 packets transmitted, 0 received, +250 errors, 100% packet loss, time 4083ms
```

4)

A) Prevent ping of death attacks on local network

Sol: (Discard all ping packets which request over 4 times per second after logging)

- 1) iptables -N PING_OF_DEATH
- 2) iptables -A PING_OF_DEATH -m limit --limit 1/s --limit-burst 4 -j ACCEPT
- 3) iptables -A PING_OF_DEATH -j LOG --log-level debug --log-prefix '[PING OF DEATH]: '
- 4) iptables -A PING_OF_DEATH -j DROP
- 5) iptables -A INPUT -p icmp --icmp-type echo-request -j PING_OF_DEATH

B) IP spoofing

Sol:

Let us assume the spoof_ips are : 127.0.0.0/8, 10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16.

Drop all spoofed

- 1) iptables -N IP_SPOOFING
- 2) iptables -A IP_SPOOFING -j LOG --log-level debug --log-prefix '[IP SPOOFING]: '
- 3) iptables -A IP_SPOOFING -j DROP
- 4) iptables -A INPUT -i eth0 -s 127.0.0.0/8 -j IP_SPOOFING
- 5) iptables -A INPUT -i eth0 -s 10.0.0.0/8 -j IP_SPOOFING
- 6) iptables -A INPUT -i eth0 -s 172.16.0.0/12 -j IP_SPOOFING
- 7) iptables -A INPUT -i eth0 -s 192.168.0.0/16 -j IP_SPOOFING

Drop packet that claiming from our own server on WAN port
8) iptables -A INPUT -i eth0 -s 10.1.37.18 -j IP_SPOOFING

Drop packet that claiming from our own internal LAN on WAN port
9) iptables -A INPUT -i eth0 -s 192.168.1.0/24 -j IP_SPOOFING

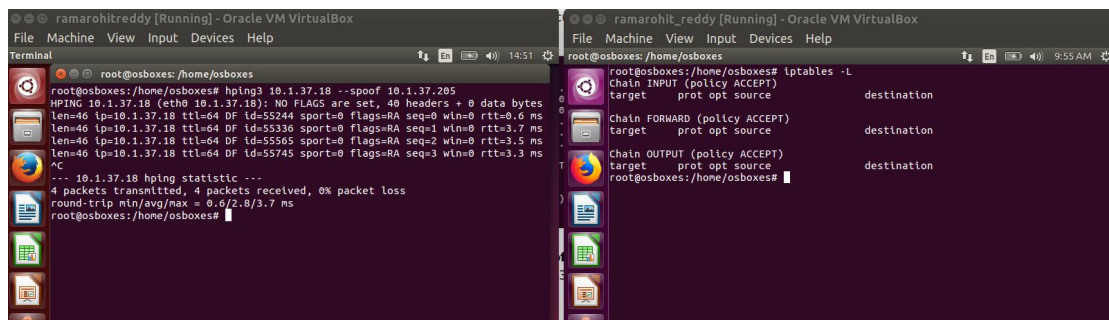
```
root@osboxes:/home/osboxes# iptables -n -L
Chain INPUT (policy ACCEPT)
target     prot opt source                destination
DROP       all  --  127.0.0.0/8            0.0.0.0/0
DROP       all  --  10.0.0.0/8            0.0.0.0/0
DROP       all  --  172.16.0.0/12         0.0.0.0/0
DROP       all  --  192.168.0.0/16        0.0.0.0/0
DROP       all  --  10.1.37.18            0.0.0.0/0
DROP       all  --  192.168.1.0/24        0.0.0.0/0

Chain FORWARD (policy ACCEPT)
target     prot opt source                destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source                destination
root@osboxes:/home/osboxes#
```

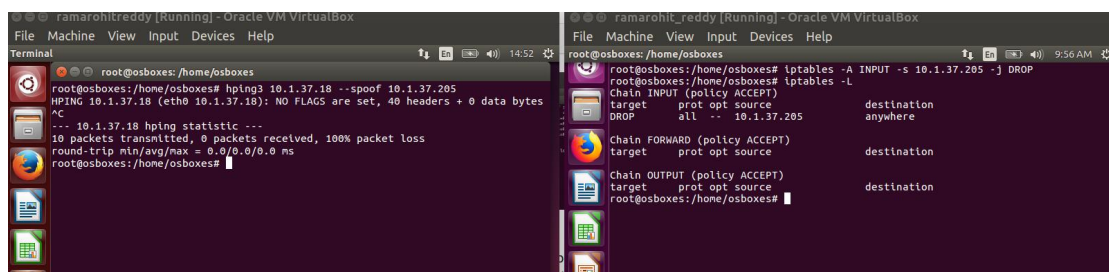
Effect of adding rules:
Client spoof itself as 10.1.37.205 using iptables command "hping3 10.1.37.18 --spoof 10.1.37.205"

Before adding:



After adding:

Now it cannot ping as ip spoofing is blocked.
But using iptables it cannot find spoofing address



The Hackers count on us being clueless about ports. Hoping we've left something "listening," they experimentally send code to our network addressed to ports we never thought of (such as port 31337, because in the

dyslexic nomenclature of script kiddies, the numbers look like EIEET -- as in, "elite" hacker). Researchers have posted several lists of ports that hackers consistently abuse. Search for such lists and consult them for real help when we interpret our firewall logs. If we leave ports open, our network could accept whatever a hacker sends. Our goal is to block every port we can. Managing our firewall largely means playing around with ports and services, blocking whole ranges of ports -- everything that our business does not require open.