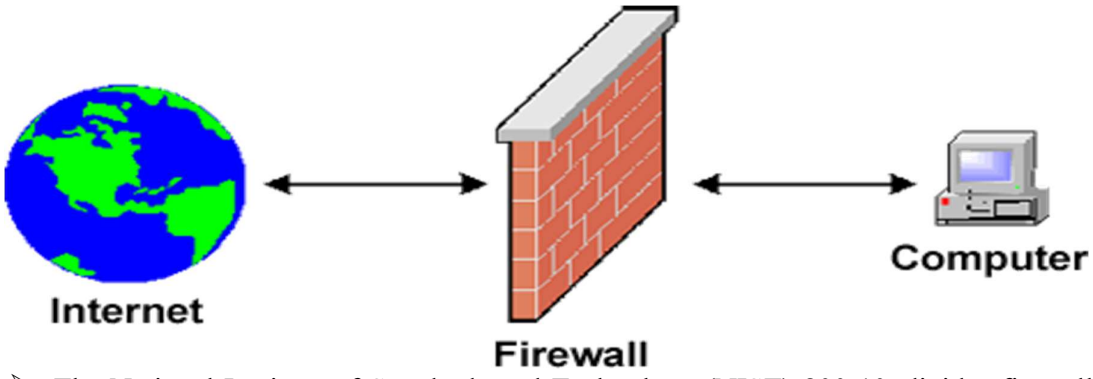
Practical – 4

Aim: Understand the concept of firewall and configure the State full Packet Inspection (SPI) firewall IPTABLES.

Firewall:

- A firewall is a network security system, either hardware or software based, that controls incoming and outgoing network traffic based on a set of rules. Acting as a barrier between a trusted network and other un-trusted networks such as the Internet or less-trusted networks such as a retail merchant's network outside of a cardholder data environment a firewall controls access to the resources of a network through a positive control model. This means that the only traffic allowed onto the network defined in the firewall policies all other traffic is denied.



- The National Institute of Standards and Technology (NIST) 800-10 divides firewalls into three basic types:

1. Packet filters
2. Stateful inspection
3. Proxys

1. Packet filters

The earliest firewalls functioned as packet filters, inspecting the packets that are transferred between computers on the Internet. When a packet passes through a packet-filter firewall, its source and destination address, protocol, and destination port number are checked against the firewall's rule set. Any packets that aren't specifically allowed onto the network are dropped (i.e., not forwarded to their destination). For example, if a firewall is configured with a rule to block Telnet access, then the firewall will drop packets destined for TCP port number 23, the port where a Telnet server application would be listening.

Packet-filter firewalls work mainly on the first three layers of the OSI reference model (physical, data-link and network), although the transport layer is used to obtain the source and destination port numbers. While generally fast and efficient, they have no ability to tell whether a packet is part of an existing stream of traffic. Because they treat each packet in isolation, this makes them vulnerable to spoofing attacks and also limits their ability to make more complex decisions based on what stage communications between hosts are at.

2. Stateful firewalls

In order to recognize a packet's connection state, a firewall needs to record all connections passing through it to ensure it has enough information to assess whether a packet is the start of a new connection, a part of an existing connection, or not part of any connection. This is what's called "stateful packet inspection." Stateful inspection was first introduced in 1994 by Check Point Software in its FireWall-1 software firewall, and by the late 1990s, it was a common firewall product feature. This additional information can be used to grant or reject access based on the packet's history in the state table, and to speed up packet processing; that way, packets that are part of an existing connection based on the firewall's state table can be allowed through without further analysis. If a packet does not match an existing connection, it's evaluated according to the rule set for new connections.

3. Proxy firewalls

Firewall proxy servers also operate at the firewall's application layer, acting as an intermediary for requests from one network to another for a specific network application. A proxy firewall prevents direct connections between either sides of the firewall; both sides are forced to conduct the session through the proxy, which can block or allow traffic based on its rule set. A proxy service must be run for each type of Internet application the firewall will support, such as an HTTP proxy for Web services.

- **IP Tables**

IPtables is an extremely flexible firewall utility built for Linux operating systems. IPtables is a command-line firewall utility that uses policy chains to allow or block traffic. When a connection tries to establish itself on our system, iptables looks for a rule in its list to match it to. If it doesn't find one, it resorts to the default action.

- **Types of Chains**

IPtables uses three different chains: input, forward, and output.

- Input: This chain is used to control the behavior for incoming connections. For example, if a user attempts to SSH into your PC/server, iptables will attempt to match the IP address and port to a rule in the input chain.
- Forward: This chain is used for incoming connections that aren't actually being delivered locally. Think of a router – data is always being sent to it but rarely actually destined for the router itself; the data is just forwarded to its target. Unless you're doing some kind of routing, NATing, or something else on your system that requires forwarding, you won't even use this chain.
- Output: This chain is used for outgoing connections. For example, if you try to ping howtogeek.com, iptables will check its output chain to see what the rules are regarding ping and howtogeek.com before making a decision to allow or deny the connection attempt.

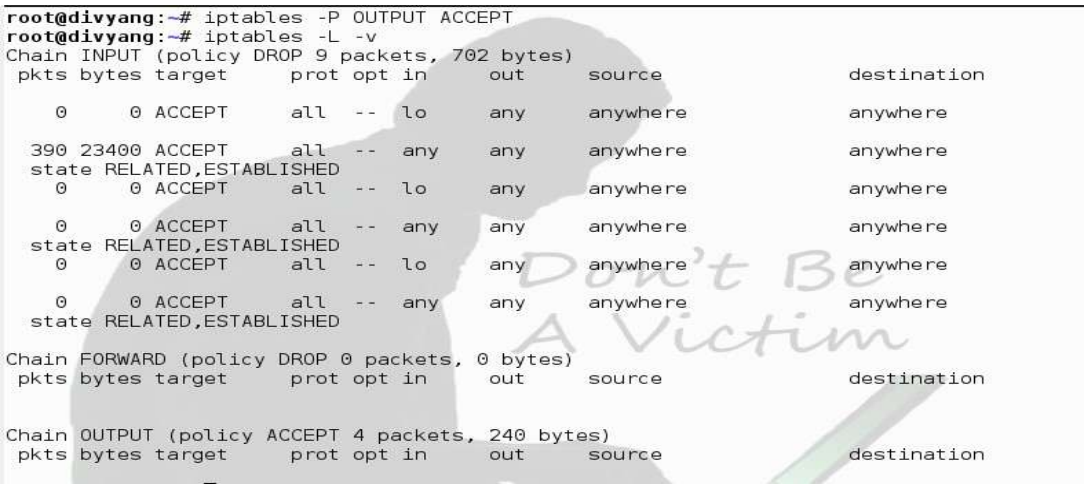
- To see if iptables is running

  iptables -L

  It is list the rules in chain or all chains.

➢ **Setup a SPI firewall that:**

1. Allow all outgoing connection
2. Block all unwanted incoming connection



iptables –F : switch to flush all existing rules so we start with a clean state from which to add new rules

iptables -P INPUT DROP : -P switch sets the default policy on the specified chain which sets the default policy on the INPUT table to drop. If an incoming packet does not match one of the following rules it will be dropped. iptables –P FORWARD DROP : set the default policy on the FORWARED chain to DROP since we're not using our computer as a router, there should not be any packets passing through our computer

iptables -P OUTPUT ACCEPT : set the default policy on the OUTPUT chain to accept. This allow outgoing traffic

iptables –A INPUT –i lo –j ACCEPT : -A switch to append (or add) a rule to specific chain, the INPUT chain in this instance. –i switch (for interface) to specify Packed matching or destined for the lo(or localhost, 127.0.0.1) interface –j (jump) to the target action for packet maching the rule – in case ACCEPT.

- Allow incoming only from one IP

```
root@divyang:~# iptables -A INPUT -s 10.10.10.130 -j ACCEPT
root@divyang:~# iptables -L -v
Chain INPUT (policy ACCEPT 217 packets, 13828 bytes)
 pkts bytes target     prot opt in     out     source               destination

    6   360 ACCEPT     all  --  any    any     10.10.10.130         anywhere


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


Chain OUTPUT (policy ACCEPT 389 packets, 23340 bytes)
 pkts bytes target     prot opt in     out     source               destination

root@divyang:~#
```

- iptables –A INPUT –s 10.10.10.130 –j ACCEPT

```
root@divyang:~# iptables -A INPUT -s 10.10.10.130 -j ACCEPT
root@divyang:~# iptables -L -v
Chain INPUT (policy ACCEPT 217 packets, 13828 bytes)
 pkts bytes target     prot opt in     out     source               destination

    6   360 ACCEPT     all  --  any    any     10.10.10.130         anywhere


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


Chain OUTPUT (policy ACCEPT 389 packets, 23340 bytes)
 pkts bytes target     prot opt in     out     source               destination

root@divyang:~#
```

```
root@divyang:~# iptables -A INPUT -s 10.10.10.127 -j ACCEPT
root@divyang:~# iptables -A INPUT -s 10.10.10.131 -j ACCEPT
root@divyang:~# iptables -P INPUT DROP
root@divyang:~# iptables -L -v
Chain INPUT (policy DROP 1 packets, 28 bytes)
 pkts bytes target     prot opt in     out     source               destinatio
n
    0     0 ACCEPT     all  --  any    any     10.10.10.127         anywhere

    0     0 ACCEPT     all  --  any    any     10.10.10.131         anywhere


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

Chain OUTPUT (policy ACCEPT 44 packets, 3132 bytes)
 pkts bytes target     prot opt in     out     source               destinatio
n
root@divyang:~#
```

After applying rules result from Different Machines are as bellow:

```
Ethernet adapter Local Area Connection:

        Connection-specific DNS Suffix  . : 220.224.142.229
        IP Address. . . . . . . . . . . . : 10.10.10.127
        Subnet Mask . . . . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . . . . : 10.10.10.10

C:\Documents and Settings\dp>ping 10.10.10.135

Pinging 10.10.10.135 with 32 bytes of data:

Reply from 10.10.10.135: bytes=32 time<1ms TTL=64
Reply from 10.10.10.135: bytes=32 time<1ms TTL=64
Reply from 10.10.10.135: bytes=32 time<1ms TTL=64
Reply from 10.10.10.135: bytes=32 time<1ms TTL=64

Ping statistics for 10.10.10.135:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

```
Ethernet adapter Local Area Connection:

   Connection-specific DNS Suffix  . : 220.224.142.229
   IPv6 Address. . . . . . . . . . . : fda3:963c:545e:0:d156:62da:6400:c81f
   Temporary IPv6 Address. . . . . . : fda3:963c:545e:0:d184:a26d:3e9b:14de
   Link-local IPv6 Address . . . . . : fe80::d156:62da:6400:c81f%11
   IPv4 Address. . . . . . . . . . . : 10.10.10.131
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 10.10.10.10

Tunnel adapter isatap.220.224.142.229:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . : 220.224.142.229

C:\Users\dp>ping 10.10.10.135

Pinging 10.10.10.135 with 32 bytes of data:
Reply from 10.10.10.135: bytes=32 time=1ms TTL=64
Reply from 10.10.10.135: bytes=32 time<1ms TTL=64
Reply from 10.10.10.135: bytes=32 time<1ms TTL=64
Reply from 10.10.10.135: bytes=32 time<1ms TTL=64

Ping statistics for 10.10.10.135:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms
```

From Another system which are not listed in Rules

```
C:\Users\dp>ping 10.10.10.135

Pinging 10.10.10.135 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 10.10.10.135:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

- iptables –A INPUT –s 10.10.10.100/24 –j ACCEPT(For whole Subnet)

```
root@divyang:~# iptables -A INPUT -s 10.10.10.100/24 -j ACCEPT
root@divyang:~# iptables -L -v
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destinatio
n
    0     0 ACCEPT     all  --  any    any     10.10.10.0/24        anywhere


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

Chain OUTPUT (policy ACCEPT 71 packets, 5244 bytes)
 pkts bytes target     prot opt in     out     source               destinatio
n
root@divyang:~#
```

- **Accept packet from trusted IP address with MAC**

Rules for Accept packet from trusted IP address with MAC are described in image.

```
root@divyang:~# iptables -A INPUT -s 10.10.10.127 -i eth1 -m mac --mac 18:AB:C0:03:18:11 -j ACCEPT
root@divyang:~# iptables -A INPUT -s 10.10.10.131 -i eth1 -m mac --mac 18:AB:C0:03:18:13 -j ACCEPT
root@divyang:~# iptables -P INPUT DROP
root@divyang:~# iptables -L -v
Chain INPUT (policy DROP 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 ACCEPT     all  --  eth1   any     10.10.10.127         anywhere            MAC 18:AB:C0:03:18:11
    0     0 ACCEPT     all  --  eth1   any     10.10.10.131         anywhere            MAC 18:AB:C0:03:18:13

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

Chain OUTPUT (policy ACCEPT 92 packets, 6645 bytes)
 pkts bytes target     prot opt in     out     source               destination
root@divyang:~#
```

After applying rules we got the following output.

1. With Same ip and mac address

```
        Physical Address. . . . . . . . . : 18-AB-C0-03-18-11
        Dhcp Enabled. . . . . . . . . . . : Yes
        Autoconfiguration Enabled . . . . : Yes
        IP Address. . . . . . . . . . . . : 10.10.10.127
        Subnet Mask . . . . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . . . . : 10.10.10.10
        DHCP Server . . . . . . . . . . . : 10.10.10.10
        DNS Servers . . . . . . . . . . . : 115.254.108.244
                                            124.124.204.36
                                            10.10.10.10
        Lease Obtained. . . . . . . . . . : Friday, August 14, 2015 3:26:22 AM
        Lease Expires . . . . . . . . . . : Saturday, August 15, 2015 3:26:22 AM

C:\Documents and Settings\dp>ping 10.10.10.135

Pinging 10.10.10.135 with 32 bytes of data:

Reply from 10.10.10.135: bytes=32 time=1ms TTL=64
Reply from 10.10.10.135: bytes=32 time=1ms TTL=64
Reply from 10.10.10.135: bytes=32 time=1ms TTL=64
Reply from 10.10.10.135: bytes=32 time=1ms TTL=64

Ping statistics for 10.10.10.135:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 1ms, Average = 1ms
```

2. Same IP but Different mac Address.

```
   Description . . . . . . . . . . . : Intel(R) PRO/1000 MT Desktop Adapter
   Physical Address. . . . . . . . . : 18-AB-C0-03-18-12
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
   IPv4 Address. . . . . . . . . . . : 10.10.10.131(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Lease Obtained. . . . . . . . . . : 14 August 2015 04:05:44
   Lease Expires . . . . . . . . . . : 15 August 2015 04:05:44
   Default Gateway . . . . . . . . . : 10.10.10.10
   DHCP Server . . . . . . . . . . . : 10.10.10.10
   DNS Servers . . . . . . . . . . . : 124.124.204.36
                                       115.254.108.244
                                       10.10.10.10
   NetBIOS over Tcpip. . . . . . . . : Enabled

Tunnel adapter isatap.220.224.142.229:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . : 220.224.142.229
   Description . . . . . . . . . . . : Microsoft ISATAP Adapter
   Physical Address. . . . . . . . . : 00-00-00-00-00-00-00-E0
   DHCP Enabled. . . . . . . . . . . : No
   Autoconfiguration Enabled . . . . : Yes

C:\Users\dp>ping 10.10.10.135

Pinging 10.10.10.135 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 10.10.10.135:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

- **Port Address Filtering**

Single port

iptables –A INPUT –p tcp --dport 21 –j ACCEPT we can also set rules for Prot Range by applying

iptables –A INPUT –p tcp - -dport 6881:6890 –j ACCEPT

```
root@divyang:~# iptables -A INPUT -p tcp --dport 21 -j ACCEPT
root@divyang:~# iptables -P INPUT DROP
root@divyang:~# iptables -L -v
Chain INPUT (policy DROP 12 packets, 936 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 ACCEPT     tcp  --  any    any     anywhere             anywhere            tcp dpt:ftp

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

Chain OUTPUT (policy ACCEPT 132 packets, 9177 bytes)
 pkts bytes target     prot opt in     out     source               destination
root@divyang:~#
```

- **LAB Assignments:**

1. Block ICMP ping using OUTPUT and echo-reply

Solution:

```
root@divyang:~# iptables -A OUTPUT -p icmp --icmp-type echo-reply -s 10.10.10.135 -d 10.10.10.131  -j DROP
root@divyang:~# iptables -L -v
Chain INPUT (policy ACCEPT 77 packets, 5834 bytes)
 pkts bytes target     prot opt in     out     source               destination

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

Chain OUTPUT (policy ACCEPT 276 packets, 20331 bytes)
 pkts bytes target     prot opt in     out     source               destination
    3   180 DROP       icmp --  any    any     10.10.10.135         10.10.10.131        icmp echo-reply
root@divyang:~#
```

After applying the rules result are as bellow.

1. From given Destination IP

```
Ethernet adapter Local Area Connection:

   Connection-specific DNS Suffix  . : 220.224.142.229
   IPv4 Address. . . . . . . . . . . : 10.10.10.131
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 10.10.10.10

Tunnel adapter isatap.220.224.142.229:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . : 220.224.142.229

C:\Users\dp>ping 10.10.10.135

Pinging 10.10.10.135 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 10.10.10.135:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

2. From another IP

```
        Connection-specific DNS Suffix  . : 220.224.142.229
        IP Address. . . . . . . . . . . . : 10.10.10.127
        Subnet Mask . . . . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . . . . : 10.10.10.10

C:\Documents and Settings\dp>ping 10.10.10.135

Pinging 10.10.10.135 with 32 bytes of data:

Reply from 10.10.10.135: bytes=32 time<1ms TTL=64
Reply from 10.10.10.135: bytes=32 time<1ms TTL=64
Reply from 10.10.10.135: bytes=32 time<1ms TTL=64
Reply from 10.10.10.135: bytes=32 time=1ms TTL=64

Ping statistics for 10.10.10.135:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms
```

2. Block ICMP ping using INPUT and echo-request

Solution 1

```
root@divyang:~# iptables -A INPUT -p icmp --icmp-type echo-reply -s 10.10.10.127 -d 10.10.10.135  -j DROP
root@divyang:~# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       icmp --  10.10.10.127        10.10.10.135        icmp echo-reply

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
root@divyang:~#
```

Solution 2

```
root@divyang:~# iptables -A INPUT -p icmp --icmp-type echo-reply -j DROP
root@divyang:~# iptables -L -v
Chain INPUT (policy ACCEPT 620 packets, 45190 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 DROP       icmp --  any    any     anywhere             anywhere            icmp echo-reply

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

Chain OUTPUT (policy ACCEPT 776 packets, 53243 bytes)
 pkts bytes target     prot opt in     out     source               destination
root@divyang:~#
```

After applying the rules result are as bellow.

For Solution 1

1. from another Machine.

```
C:\Users\dp>ping 10.10.10.135

Pinging 10.10.10.135 with 32 bytes of data:
Reply from 10.10.10.135: bytes=32 time<1ms TTL=64
Reply from 10.10.10.135: bytes=32 time=1ms TTL=64
Reply from 10.10.10.135: bytes=32 time=1ms TTL=64
Reply from 10.10.10.135: bytes=32 time=1ms TTL=64

Ping statistics for 10.10.10.135:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\Users\dp>
```

2. From Linux Machine on which firewall rules applie

```
root@divyang:~# ping 10.10.10.127
PING 10.10.10.127 (10.10.10.127) 56(84) bytes of data.
^Z
[1]+  Stopped                 ping 10.10.10.127
root@divyang:~#
```

```
root@divyang:~# ping 10.10.10.131
PING 10.10.10.131 (10.10.10.131) 56(84) bytes of data.
64 bytes from 10.10.10.131: icmp_seq=1 ttl=128 time=1.55 ms
64 bytes from 10.10.10.131: icmp_seq=2 ttl=128 time=1.80 ms
64 bytes from 10.10.10.131: icmp_seq=3 ttl=128 time=1.65 ms
64 bytes from 10.10.10.131: icmp_seq=4 ttl=128 time=1.66 ms
^Z
[1]+  Stopped                 ping 10.10.10.131
root@divyang:~#
```

For Solution 2

```
root@divyang:~# ping 10.10.10.131
PING 10.10.10.131 (10.10.10.131) 56(84) bytes of data.
^Z
[1]+  Stopped                 ping 10.10.10.131
root@divyang:~# ping 10.10.10.127
PING 10.10.10.127 (10.10.10.127) 56(84) bytes of data.
^Z
[2]+  Stopped                 ping 10.10.10.127
root@divyang:~#
```

3. Block FTP using OUTPUT or INPUT (allow ftp server for your subnet only)

Solution 1(For INPUT)

```
root@divyang:~# iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
root@divyang:~# iptables -A OUTPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
root@divyang:~# iptables -A INPUT -p tcp --dport 21 -s 10.10.10.10/24 -m state --state NEW -j ACCEPT
root@divyang:~# iptables -L -v
Chain INPUT (policy ACCEPT 796 packets, 56561 bytes)
 pkts bytes target     prot opt in     out     source               destination
  123  7380 ACCEPT     all  --  any    any     anywhere             anywhere            state RELATED,ESTABLISH
ED
    0     0 ACCEPT     tcp  --  any    any     10.10.10.0/24        anywhere            tcp dpt:ftp state NEW

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

Chain OUTPUT (policy ACCEPT 938 packets, 63083 bytes)
 pkts bytes target     prot opt in     out     source               destination
  111  6660 ACCEPT     all  --  any    any     anywhere             anywhere            state RELATED,ESTABLISH
ED
root@divyang:~#
```

After applying the rules result are as bellow.

```
root@divyang:~# ftp 10.10.10.127
Connected to 10.10.10.127.
220-Microsoft FTP Service
220 Hi..FTP BY DIVYANG ON XP Server.
Name (10.10.10.127:root): dp
331 Password required for dp.
Password:
230-WELCOME..
230 User dp logged in.
Remote system type is Windows_NT.
ftp>
```

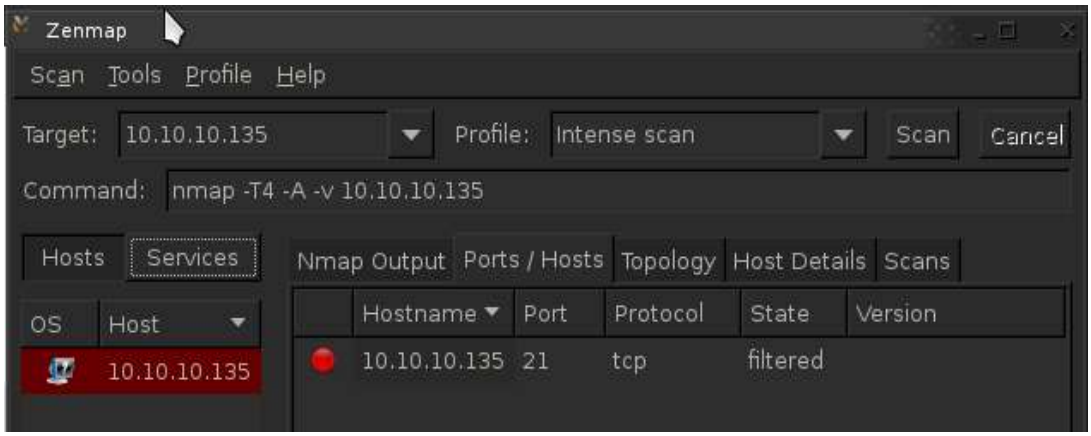
Solution 2 (For OUTPUT)

```
root@divyang:~# iptables -F
root@divyang:~# iptables -A OUTPUT -p tcp --dport 21 -s 10.10.10.10/24 -m state --state NEW -j DROP
root@divyang:~# iptables -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
DROP       tcp  --  10.10.10.0/24        anywhere            tcp dpt:ftp state NEW
root@divyang:~#
```

Result after applying the Rules.

```
root@divyang:~# ftp 10.10.10.127
ftp: connect: Connection timed out
ftp>
```

Checked the port using Zen map tools



4. **ALLOW ssh using INPUT**
   Solution

```
root@divyang:~# iptables -A INPUT -p tcp --dport 22 -j ACCEPT
root@divyang:~# iptables -P INPUT DROP
root@divyang:~# iptables -L -v
Chain INPUT (policy DROP 2 packets, 120 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 ACCEPT     tcp  --  any    any     anywhere             anywhere            tcp dpt:ssh

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

Chain OUTPUT (policy ACCEPT 1671 packets, 107K bytes)
 pkts bytes target     prot opt in     out     source               destination
root@divyang:~#
```

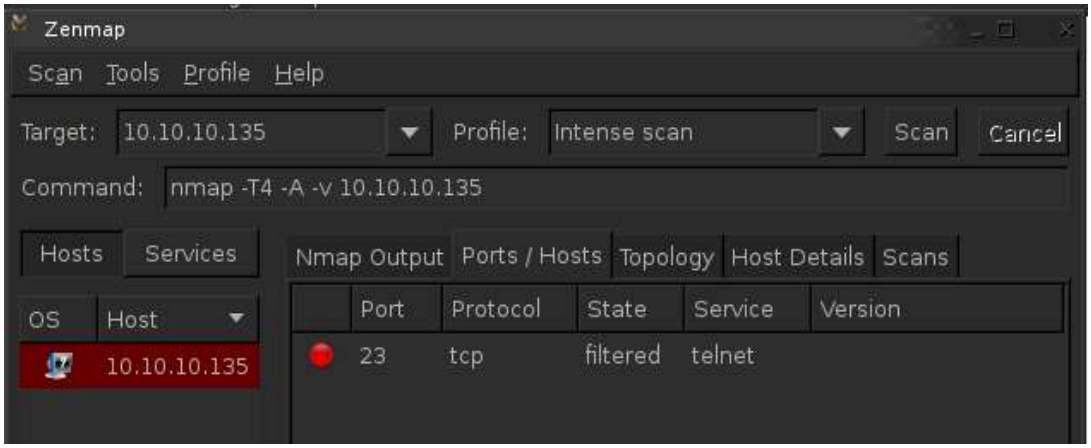
5. **Block TELNET using OUTPUT and INPUT.**
   Solution 1 (For INPUT)

```
root@divyang:~# iptables -F
root@divyang:~# iptables -A INPUT -p tcp --dport 23 -j DROP
root@divyang:~# iptables -L -v
Chain INPUT (policy ACCEPT 43 packets, 2548 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 DROP       tcp  --  any    any     anywhere             anywhere            tcp dpt:t
elnet

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

Chain OUTPUT (policy ACCEPT 42 packets, 2520 bytes)
 pkts bytes target     prot opt in     out     source               destination
root@divyang:~#
```

Solution 1 (For OUTPUT)

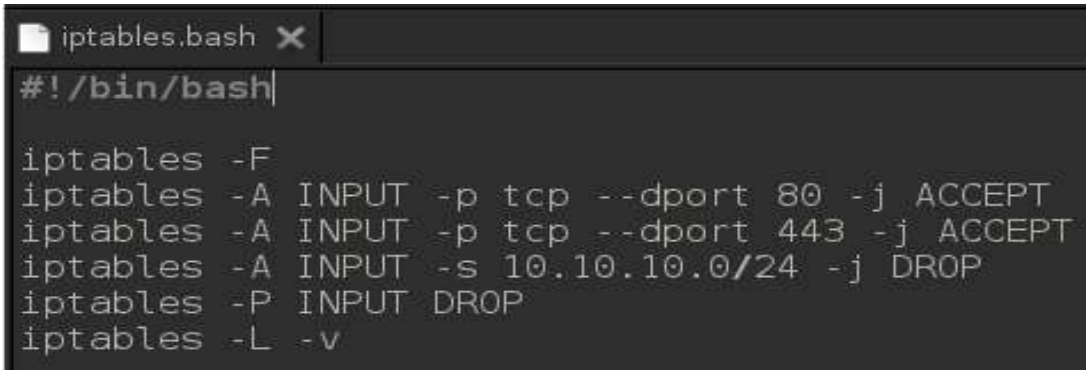
```
root@divyang:~# iptables -F
root@divyang:~# iptables -A OUTPUT -p tcp --sport 23 -j DROP
root@divyang:~# iptables -L -v
Chain INPUT (policy ACCEPT 138 packets, 8270 bytes)
 pkts bytes target     prot opt in     out     source               destination

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

Chain OUTPUT (policy ACCEPT 133 packets, 7980 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 DROP       tcp  --  any    any     anywhere             anywhere            tcp spt:t
elnet
root@divyang:~#
```

After applying the rules result are as bellow.



6. **Allow web server only to outside world.**

Solution using bash file



After applying the rules result are as bellow.