Beginner Guide to IPtables

posted in PENETRATION TESTING on FEBRUARY 9, 2018 by RAJ CHANDEL SHARE

Hello friends!! In this article we are going to discuss on Iptables and its uses. **Iptables** is a command-line firewall, installed by default on all official Ubuntu distributions. Using Iptables, you can label a set of rules, that will be go after by the Linux kernel to verify all incoming and outgoing network traffic.

Today we will look at some basic concept of Ipatble using various Iptables options to generate a **Filter Table** which will filter the incoming and outgoing traffic

Basic Iptables Options

-A : Add this rule to a rule chain.

-L: List the current filter rules.

-m conntrack : Allow filter rules to match based on connection state. Permits the use of the –ctstate option.

–ctstate: Define the list of states for the rule to match on. Valid states are:

- **NEW** – The connection has not yet been seen.
- **RELATED** – The connection is new, but is related to another connection already permitted.
- **ESTABLISHED** – The connection is already established.
- **INVALID** – The traffic couldn't be identified for some reason.

-m limit: Require the rule to match only a limited number of times. Allows the use of the –limit option.

Useful for limiting logging rules:

- –limit – The maximum matching rate, given as a number followed by "/second", "/minute", "/hour", or "/day" depending on how often you want the rule to match. If this option is not used and -m limit is used, the default is "3/hour".

-p: Describe the connection protocol used.

–dport : The destination port(s) required for this rule. A single port may be given, or a range may be given as start: end, which will match all ports from start to end, inclusive.

-j : Jump to the specified target. By default, iptables allows four targets:

- **ACCEPT** – Accept the packet and stop processing rules in this chain.
- **REJECT**– Reject the packet and notify the sender that we did so, and stop processing rules in this chain.
- **DROP**– Silently ignore the packet, and stop processing rules in this chain.

- **LOG**– Log

-I: Inserts a rule. Takes two options, the chain to insert the rule into, and the rule number it should be.

-I: INPUT 5 would insert the rule into the INPUT chain and make it the 5th rule in the list.

-s: –source – address [/mask] source specification

-d: –destination – address[/mask] destination specification

Iptables follow Ipchain rules which is nothing but the bunch of firewall rules to control incoming and outgoing traffic

Three Important Types Iptable chains

Input Chain: Input chain rule rule is used to manage the activities of incoming traffic towards server.

Output Chain: Ouput chain rule is used to manage the activities of outgoing traffic from your server.

Forward Chain: A forward chain rule is used for adding up rules related to forwarding of an ip packet. This is usually used while you have a Linux machine as router linking two networks collectively.

As discribed above by default install iptable is availabe in all Ubuntu distribution but if it is not installed in any Linux based system and you want to install it then excute given below command.

sudo apt-get install iptables

By default iptable is blank which allows all incoming and outgoing connection traffic without filtering them. In order to verify inbuilt rules of iptable we need to execute following command which displays the list of rules if added in iptables.

Categories

sudo iptables -L -v

here **-L** is used for display the chain rules of iptables and **-v** for complete information.

```
pentest@ubuntu:~$ sudo iptables -L -v
Chain INPUT (policy ACCEPT 3 packets, 132 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 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
```

Allow Incoming Traffic

In order to allow traffic for any particular port you can use given below command here we have accept incoming on port 22 for SSH, 80 for HTTP and 443for HTTPS respectively

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

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

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

So it will allow tcp connection when traffic will coming on port 22, 80 and 443.

```
pentest@ubuntu:~$ sudo iptables -A INPUT -p tcp --dport 22 -j ACCEPT
pentest@ubuntu:~$ sudo iptables -A INPUT -p tcp --dport 80 -j ACCEPT
pentest@ubuntu:~$ sudo iptables -A INPUT -p tcp --dport 443 -j ACCEPT
```

Drop/Deny Incoming Traffic

In order to deny traffic for any particular port you can use given below command here we have drop incoming on port 21 for FTP and 23 for Telnet respectively

sudo iptables -A INPUT -p tcp –dport 21 -j DROP

sudo iptables -A INPUT -p tcp –dport 23 -j DROP

So it will deny tcp connection when traffic will coming on port 21, 23 and give a message **Time Out**.

```
pentest@ubuntu:~$ sudo iptables -A INPUT -p tcp --dport 21 -j DROP
pentest@ubuntu:~$ sudo iptables -A INPUT -p tcp --dport 23 -j DROP
```

Reject Incoming Traffic

Reject and Drop action closely work same in order to obstruct the incoming traffic from establishing connection with server only the difference is that, here it will send message with "**ICMP message Port Unreachable**" and drop the incoming packet. You can use given below command here we have reject incoming on port 25 for SMTP.

sudo iptables -A INPUT -p tcp –dport 25 –j REJECT

So it will drop tcp connection when traffic will coming on port 25 and give a message **Destination Port unreachable**.

```
pentest@ubuntu:~$ sudo iptables -A INPUT -p tcp --dport 25 -j REJECT
```

Allow Incoming Traffic from Specific IP

In order to allow traffic form only a particular IP to establish a secure connection between server and client you can execute given below command

sudo iptables -A INPUT -s 192.168.1.104 -j ACCEPT

It will accept packet coming from network 192.168.1.104

```
pentest@ubuntu:~$ sudo iptables -A INPUT -s 192.168.1.104 -j ACCEPT
pentest@ubuntu:~$
```

Block Specific Network IP

In order to deny traffic form only a particular IP to establish a secure your server from attacker's IP you can execute given below command

sudo iptables -A INPUT -s 192.168.1.102 -j DROP

It will deny packet coming from network 192.168.1.102

```
pentest@ubuntu:~$ sudo iptables -A INPUT -s 192.168.1.102 -j DROP
pentest@ubuntu:~$
```

Block Specific Network Interface

To block a specific network interface, for example eth0, execute given below command which drop the incoming traffic coming from 10.10.10.10

sudo iptables -A INPUT -i eth0 -s 10.10.10.10-j DROP

Here you can change the action to allow traffic from a particular network interface using –j ACCEPT options.

```
pentest@ubuntu:~$ sudo iptables -A INPUT -i eth0 -s 10.10.10.10 -j DROP
pentest@ubuntu:~$
```

Block Specific IP Range

To block a specific IP range in order to deny, the incoming traffic coming from specific range of IP. Execute given below command which drop incoming packet coming from IP 192.168.1.100 till IP 192.168.1.200

sudo iptables -A INPUT -m iprange –src-range 192.168.1.100-192.168.1.200 -j DROP

Here you can change the action to allow traffic from a particular IP range using –j ACCEPT options.

```
pentest@ubuntu:~$ sudo iptables -A INPUT -m iprange --src-range 192.168.1.100-192.168.1.200
 -j DROP
```

Block Specific Mac Address

To block a specific Mac address in order to deny, the incoming traffic coming from specific machine. Execute given below command which drop incoming packet coming from given Mac address or attacker machine.

sudo iptables -A INPUT -m mac –mac-source FC:AA:14:6A:9A:A2 -j DROP

Here you can change the action to allow traffic from a particular Mac address using –j ACCEPT options.

```
pentest@ubuntu:~$ sudo iptables -A INPUT -m mac --mac-source FC:AA:14:6A:9A:A2 -j DROP
```

Block Ping Request

Network administrator always concern with network security therefore they always Block Ping request either by using Drop or Reject action , here we are blocking Ping request using DROP option as given in below command.

sudo iptables -A INPUT -p icmp -i eth0 -j DROP

```
pentest@ubuntu:~$ sudo iptables -A INPUT -p icmp -i eth0 -j DROP
```

View List of Applied Chain rules

In order to view our applied chain rules once again we are going to execute given below command which will dump list of Iptable rules.

sudo iptables -L

From given below image you can observe 4 columns which contains records of IPtable rules.

Here these columns define following information:

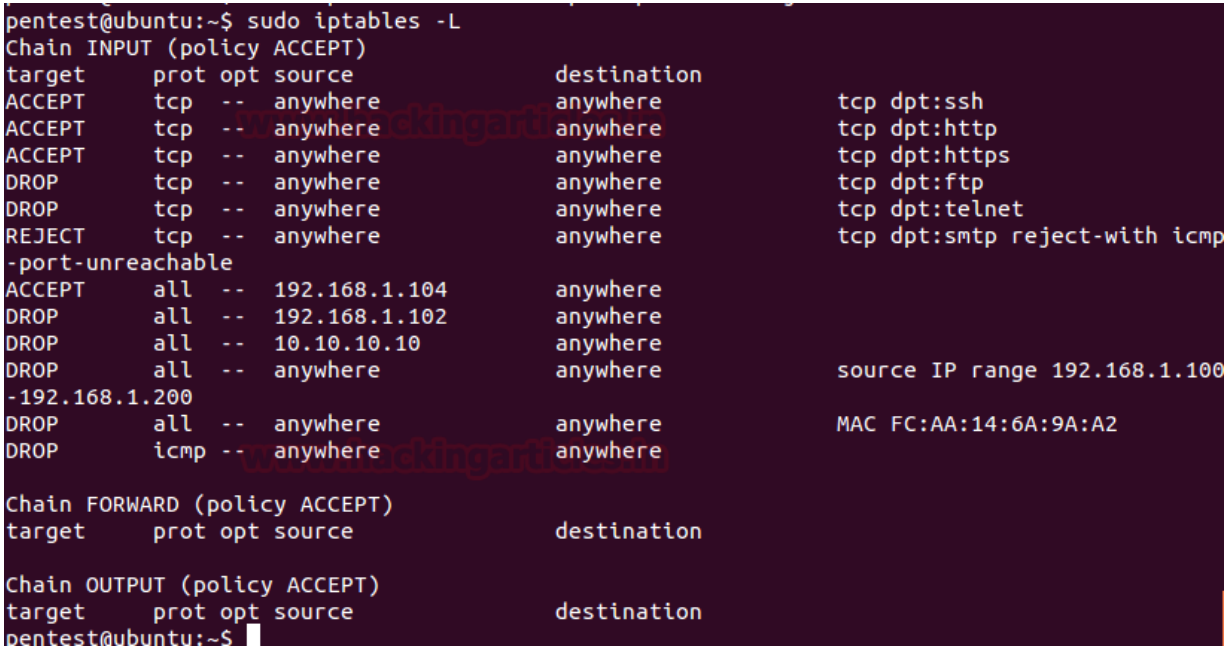
Target: Defines applied action

Prot: stand for Protocol type that can TCP, ICMP or UDP

Option: further option to define rule, here it is blank

Source: Incoming traffic network IP Address

Destination: Host IP address which will receive incoming traffic packet.



Now if someone tries to Ping the server machine as shown in given below image, so here you can read the message “Request timed out” which means the server machine has drop our ICMP request packet.

```
C:\Users\Pentest>ping 192.168.1.103

Pinging 192.168.1.103 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 192.168.1.103:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

Deleting Any Rule

In order to delete any rule of your Iptable to remove it from inside your filter table you can use **option -D** with input rule number. We are going to remove our last rule ICMP drop connection which was at number 12 in the given list of rule.

sudo iptables -D INPUT 12

Here you can replace number 12 from any other number which rule you wish to remove according to your list of rules.

```
pentest@ubuntu:~$ sudo iptables -D INPUT 12
pentest@ubuntu:~$
```

Let's view our remaining chain rules once again using -L option as done above. From given below image you can observe that now the list contain only 11 rules and eliminated rule ICMP drop the connection.

```
pentest@ubuntu:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:ssh
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:http
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:https
DROP       tcp  --  anywhere             anywhere             tcp dpt:ftp
DROP       tcp  --  anywhere             anywhere             tcp dpt:telnet
REJECT     tcp  --  anywhere             anywhere             tcp dpt:smtp reject-with icmp
-port-unreachable
ACCEPT     all  --  192.168.1.104        anywhere
DROP       all  --  192.168.1.102        anywhere
DROP       all  --  10.10.10.10          anywhere
DROP       all  --  anywhere             anywhere             source IP range 192.168.1.100
-192.168.1.200
DROP       all  --  anywhere             anywhere             MAC FC:AA:14:6A:9A:A2

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
pentest@ubuntu:~$
```

Flush IPtables

If you want to remove entire set of rule in order to flush your Iptable then use option -F to flush your ipatble applied rules and execute given below command.

sudo iptables -F

Now once again when we had viewed the list of rule, this time we got empty table as shown in given below image.

```
pentest@ubuntu:~$ sudo iptables -F
pentest@ubuntu:~$ sudo 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
pentest@ubuntu:~$
```


Source: https://help.ubuntu.com/community/IptablesHowTo

Author: AArti Singh is a Researcher and Technical Writer at Hacking Articles an Information Security Consultant Social Media Lover and Gadgets. Contact **here**

Share this:

Like this:

Loading...

ABOUT THE AUTHOR

RAJ CHANDEL

Raj Chandel is a Skilled and Passionate IT Professional especially in IT-Hacking Industry. At present other than his name he can also be called as An Ethical Hacker, A Cyber Security Expert, A Penetration Tester. With years of quality Experience in IT and software industry

PREVIOUS POST
← PAYLOAD PROCESSING RULE IN BURP SUITE (PART 2)

NEXT POST
BIND PAYLOAD USING SFX ARCHIVE WITH TROJANIZER →

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment

Name *

Email *

Website

Save my name, email, and website in this browser for the next time I comment.

POST COMMENT

Notify me of follow-up comments by email.

Notify me of new posts by email.