Cycle 8

Iptables in linux.

1. Aim: Use of iptables in linux

2. Objectives: To study how to create iptables in linux.

3. Theory:

iptables is a command line interface used to set up and maintain tables for the Netfilter firewall for IPv4, included in the Linux kernel. The firewall matches packets with rules defined in these tables and then takes the specified action on a possible match.

**Tables** is the name for a set of chains.

**Chain** is a collection of rules.

**Rule** is condition used to match packet.

**Target** is action taken when a possible rule matches. Examples of the target are ACCEPT, DROP, QUEUE.

**Policy** is the default action taken in case of no match with the inbuilt chains and can be ACCEPT or DROP.

Syntax:

iptables --table TABLE -A/-C/-D... CHAIN rule --jump Target

**TABLE**

There are five possible tables:

**filter**: Default used table for packet filtering. It includes chains like INPUT, OUTPUT and FORWARD.

**nat** : Related to Network Address Translation. It includes PREROUTING and POSTROUTING chains.

**mangle** : For specialised packet alteration. Inbuilt chains include PREROUTING and OUTPUT.

**raw** : Configures exemptions from connection tracking. Built-in chains are PREROUTING and OUTPUT.

**security** : Used for Mandatory Access Control

CHAINS

There are few built-in chains that are included in tables. They are:

* **INPUT :**set of rules for packets destined to localhost sockets.
* **FORWARD :**for packets routed through the device.
* **OUTPUT :**for locally generated packets, meant to be transmitted outside.
* **PREROUTING :**for modifying packets as they arrive.
* **POSTROUTING :**for modifying packets as they are leaving.

**OPTIONS**

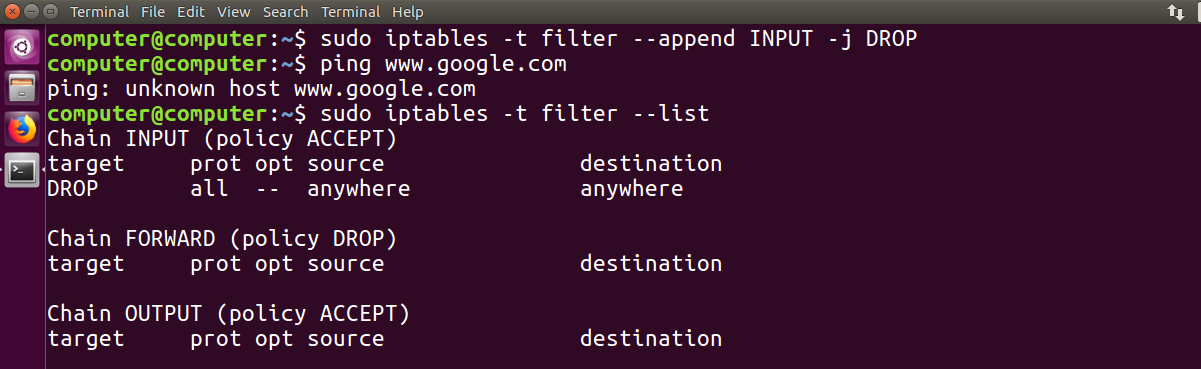
1. **-A, –append :** Append to the chain provided in parameters.

**Syntax:**

iptables [-t table] --append [chain] [parameters]

**Example:** This command drops all the traffic coming on any port.

iptables -t filter --append INPUT -j DROP

**Output:**  


1. **-D, –delete :** Delete rule from the specified chain.

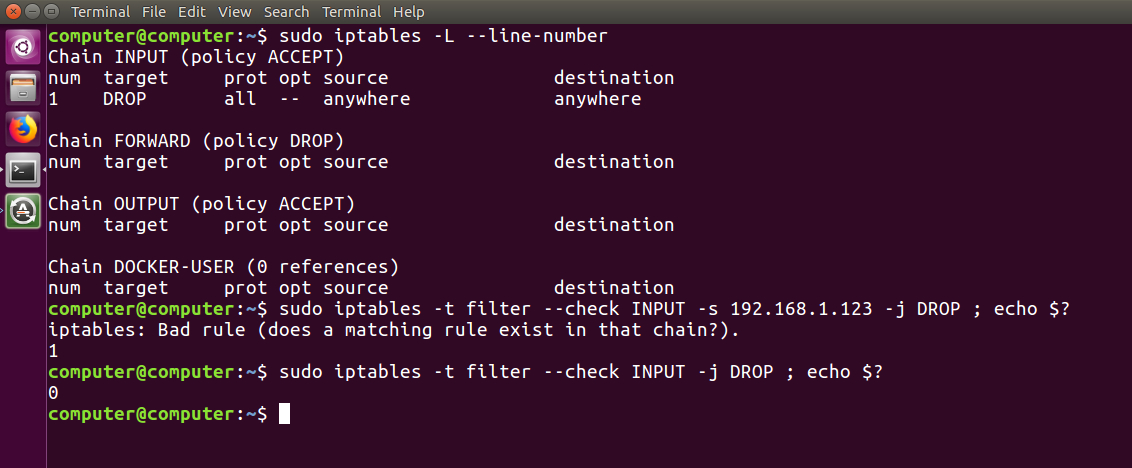
**Syntax:**

iptables [-t table] --delete [chain] [rule\_number]

**Example:** This command deletes the rule 2 from INPUT chain.

iptables -t filter --delete INPUT 2

**Output:**

1. iptables -t filter --check INPUT -s 192.168.1.123 -j DROP
2. **Output:**  
   

**PARAMETERS**

The parameters provided with the *iptables*command is used to match the packet and perform the specified action. The common parameters are:

1. **-p, –proto :** is the protocol that the packet follows. Possible values maybe: tcp, udp, icmp, ssh etc.

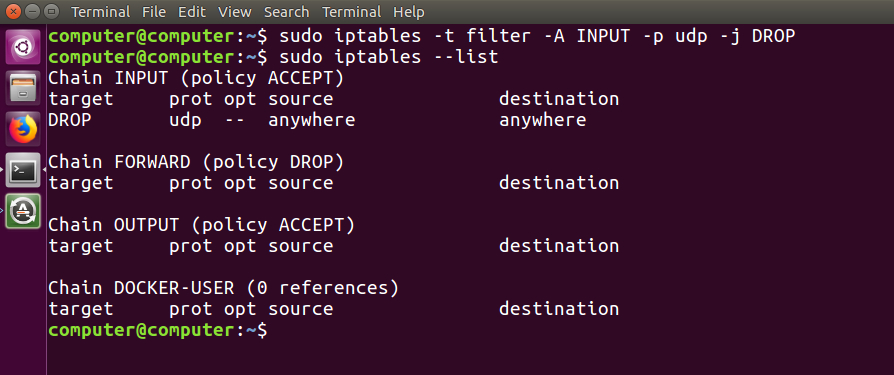
**Syntax:**

iptables [-t table] -A [chain] -p {protocol\_name} [target]

**Example:** This command appends a rule in the INPUT chain to drop all udp packets.

iptables -t filter -A INPUT -p udp -j DROP

**Output:**



1. **-s, –source:** is used to match with the source address of the packet.

**Syntax:**

iptables [-t table] -A [chain] -s {source\_address} [target]

**Example:** This command appends a rule in the INPUT chain to accept all packets originating from 192.168.1.230.

iptables -t filter -A INPUT -s 192.168.1.230 -j ACCEPT

**Output:**  


1. **-d, –destination :** is used to match with the destination address of the packet.

**Syntax:**

iptables [-t table] -A [chain] -d {destination\_address} [target]

**Example:** This command appends a rule in the OUTPUT chain to drop all packets destined for 192.168.1.123.

iptables -t filter -A OUTPUT -d 192.168.1.123 -j DROP

**Output:**  

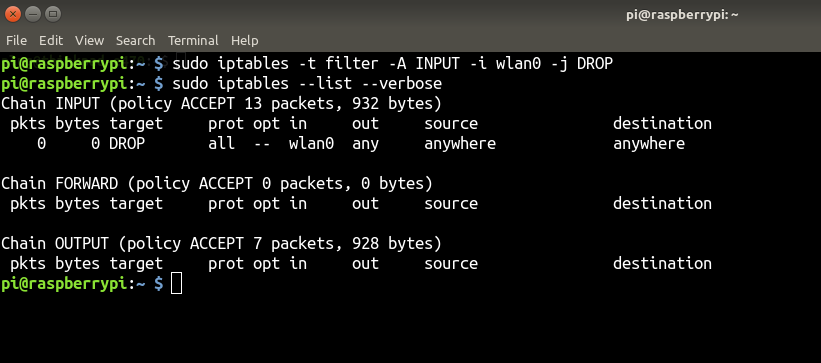

1. **-i, –in-interface :** matches packets with the specified in-interface and takes the action.

**Syntax:**

iptables [-t table] -A [chain] -i {interface} [target]

**Example:** This command appends a rule in the INPUT chain to drop all packets destined for wireless interface.

iptables -t filter -A INPUT -i wlan0 -j DROP

**Output:**  


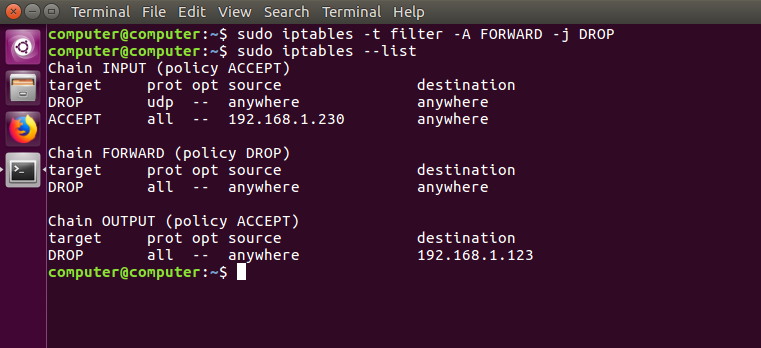
1. **-o, –out-interface :** matches packets with the specified out-interface.
2. **-j, –jump :** this parameter specifies the action to be taken on a match.

**Syntax:**

iptables [-t table] -A [chain] [parameters] -j {target}

**Example:** This command adds a rule in the FORWARD chain to drop all packets.

iptables -t filter -A FORWARD -j DROP

**Output:**  


* While trying out the commands, you can remove all filtering rules and user created chains.
* sudo iptables --flush
* To save the iptables configuration use:

sudo iptables-save

* Restoring iptables config can be done with:

sudo iptables-restore

**4. Conclusion:**There are many other firewall utilities and some that may be easier, but iptables is a good learning tool, if only because it exposes some of the underlying netfilter structure and because it is present in so many systems.