LINUX

Configure IPTables on CentOS 6

September 15, 2014, 15:22 7

IPTables is a service on linux systems, which allows a system administrator to configure rules and chains in tables provides by the Linux kernel firewall. IPTables is a Linux firewall service which enables you to accept, reject or drop (,…) packages based on the rules you applied.

In this post we will learn how to cofigure iptables for basic usage. We will learn how to **start, stop and restart the service**, where to find the **iptables configuration file**, how to **add a new rule** and how to **delete a rule** from configuration, how to **reload/restart iptables** and how to **check if rules were really applied**.

Please note that different services manage different protocols which means that IPTables service applied to IPv4 and ip6tables service applies to IPv6 protocol.

IPTables on Linux are very powerful and can provide complex routing rules. Essentially IPTables are defined by tables containing chains of rules, which are applied to packets. Five predefined chains are:

PREROUTING: Packets will enter this chain before a routing decision is made.
INPUT: Packet is going to be locally delivered. It does not have anything to do with processes having an opened socket; local delivery is controlled by the “local-delivery” routing table: ip route show table local.
FORWARD: All packets that have been routed and were not for local delivery will traverse this chain.
OUTPUT: Packets sent from the machine itself will be visiting this chain.

POSTROUTING: Routing decision has been made. Packets enter this chain just before handing them off to the hardware.

Configure IPTables on CentOS 6

Let's Learn How To Configure IPTables!

1. IPTables InitScript

IPTables service is managed **just as any other service** in CentOS 6 linux via **initscript**. As we know, init script is used to **manage (start, stop, reload,…) the iptables service** and not to configure iptables itself. The iptables **initscript is located at /etc/init.d/iptables** (ip6tables for IPv6) and accepts most of the usual parameters like start, stop, reload, restart, condrestart, status, panic and save.

- /etc/init.d/iptables start

Start is used to start the iptables service. Please be careful and check the rules in /etc/sysconfig/iptables before starting the service since this

may lead to blocking some service you do not really want! ☺

- /etc/init.d/iptables stop

  Stop is used to stop iptables service. This means no firewall rule will be applied and the machine will be accessible on all running ports.

- /etc/init.d/iptables reload

  Reload will reload the currently persistent iptables configuration from /etc/sysconfig/iptables.

- /etc/init.d/iptables condrestart

  Condrestart will restart iptables service if service is already running.

- /etc/init.d/iptables status

  Status will return currently active iptables rules. Please note this includes manually (explained in “Adding rules”) applied iptables rules.

- /etc/init.d/iptables panic

  Panic will set all chains to DROP policy. This is useful in cases when your machine is under attack. This will DROP every incomming packets, meaning it will DROP your SSH connection too!

- **/etc/init.d/iptables** save

  Save will save currently active (including manually applied rules) iptables configuration to /etc/sysconfig/iptables configuration file, making it persistent.

The following is the **default** persistent **configuration of IPTables** on the latest CentOS 6 linux:

```
[root@geekpeek1 ~]# /etc/init.d/iptables status
 Table: filter
 Chain INPUT (policy ACCEPT)
 num target prot opt source destination
 1 ACCEPT all -- 0.0.0.0/0 0.0.0.0/0 state RELATED,ESTABLISHED
 2 ACCEPT icmp -- 0.0.0.0/0 0.0.0.0/0
 3 ACCEPT all -- 0.0.0.0/0 0.0.0.0/0
 4 ACCEPT tcp -- 0.0.0.0/0 0.0.0.0/0 state NEW tcp dpt:22
 5 REJECT all -- 0.0.0.0/0 0.0.0.0/0 reject-with icmp-host-prohibited

Chain FORWARD (policy ACCEPT)
 num target prot opt source destination
 1 REJECT all -- 0.0.0.0/0 0.0.0.0/0 reject-with icmp-host-prohibited

Chain OUTPUT (policy ACCEPT)
 num target prot opt source destination
```

To add additional rules to the default configuration we need to learn how to configure iptables.

2. Adding IPTables Rules Manually (non-persistent)

We will explaing how to **add or remove an IPTables rules** manually. Maybe manually is not the right word to describe this but anyways. Let see how to configure iptables! We can add or remove iptables rules manually **using the /sbin/iptables command**.

The **syntax** is as follows:

Allowing HTTP traffic (TCP port 80)

```
[root@geekpeek1 ~]# iptables -A INPUT -p tcp -m state --state NEW -m tcp --dport 80 -j ACCEPT
```

Allowing SNMP traffic (UDP port 161)

```
[root@geekpeek1 ~]# iptables -A INPUT -p udp -m state --state NEW -m udp --dport 161 -j ACCEPT
```

Allowing HTTPS traffic from specific IP address (TCP port 443)

```
[root@geekpeek1 ~]# iptables -A INPUT -s 8.8.8.8 -p tcp -m state --state NEW -m tcp --dport 443 -
j ACCEPT
```

These are some **basic examples**, just change the ports and protocols according to your liking and also the IP address you want to allow traffic from.

We can check if everything is as we configured it with **iptables status command**:

```
[root@geekpeek1 ~]# /etc/init.d/iptables status
 Table: filter
 Chain INPUT (policy ACCEPT)
 num target prot opt source destination
 1 ACCEPT all -- 0.0.0.0/0 0.0.0.0/0 state RELATED,ESTABLISHED
 2 ACCEPT icmp -- 0.0.0.0/0 0.0.0.0/0
 3 ACCEPT all -- 0.0.0.0/0 0.0.0.0/0
 4 ACCEPT tcp -- 0.0.0.0/0 0.0.0.0/0 state NEW tcp dpt:22
 5 REJECT all -- 0.0.0.0/0 0.0.0.0/0 reject-with icmp-host-prohibited
 6 ACCEPT udp -- 0.0.0.0/0 0.0.0.0/0 state NEW udp dpt:161
```

```
 7 ACCEPT tcp -- 8.8.8.8 0.0.0.0/0 state NEW tcp dpt:443

Chain FORWARD (policy ACCEPT)
 num target prot opt source destination
 1 REJECT all -- 0.0.0.0/0 0.0.0.0/0 reject-with icmp-host-prohibited

Chain OUTPUT (policy ACCEPT)
 num target prot opt source destination
```

If everything looks OK, we **MUST NOT FORGET (!)** to **save** the IPTables rules and make it **PERSISTENT** and **RELOAD** or restart the **IPTables** configuration!

This way the current configuration is saved to /etc/sysconfig/iptables file and the service reloads it's configuration. To do this we must run the following command:

```
[root@geekpeek1 ~]# /etc/init.d/iptables save
iptables: Saving firewall rules to /etc/sysconfig/iptables:[ OK ]
[root@geekpeek1 ~]# /etc/init.d/iptables reload
iptables: Trying to reload firewall rules: [ OK ]
```

3. Removing IPTables Rules Manually (non-persistent)

Let's check **iptables status** again to see the currently applied rules and **choose the ones we want to remove**:

```
[root@geekpeek1 ~]# /etc/init.d/iptables status
 Table: filter
```

```
Chain INPUT (policy ACCEPT)
num target prot opt source destination
1 ACCEPT all -- 0.0.0.0/0 0.0.0.0/0 state RELATED,ESTABLISHED
2 ACCEPT icmp -- 0.0.0.0/0 0.0.0.0/0
3 ACCEPT all -- 0.0.0.0/0 0.0.0.0/0
4 ACCEPT tcp -- 0.0.0.0/0 0.0.0.0/0 state NEW tcp dpt:22
5 REJECT all -- 0.0.0.0/0 0.0.0.0/0 reject-with icmp-host-prohibited
6 ACCEPT udp -- 0.0.0.0/0 0.0.0.0/0 state NEW udp dpt:161
7 ACCEPT tcp -- 8.8.8.8 0.0.0.0/0 state NEW tcp dpt:443

Chain FORWARD (policy ACCEPT)
num target prot opt source destination
1 REJECT all -- 0.0.0.0/0 0.0.0.0/0 reject-with icmp-host-prohibited

Chain OUTPUT (policy ACCEPT)
num target prot opt source destination
```

The **easiest way to remove** IPTables rules manually is **by it's rule number**. Rule number is the number before the rule, we want to remove.

If you would like to remove the last rule added "*Allowing HTTPS traffic from specific IP address*" we must issue the following command:

```
[root@geekpeek1 ~]# iptables -D INPUT 7
```

Checking the **iptables status again** shows the **rule number 7 was successfully removed**:

```
[root@geekpeek1 ~]# /etc/init.d/iptables status
```

```
Table: filter
Chain INPUT (policy ACCEPT)
num target prot opt source destination
1 ACCEPT all -- 0.0.0.0/0 0.0.0.0/0 state RELATED,ESTABLISHED
2 ACCEPT icmp -- 0.0.0.0/0 0.0.0.0/0
3 ACCEPT all -- 0.0.0.0/0 0.0.0.0/0
4 ACCEPT tcp -- 0.0.0.0/0 0.0.0.0/0 state NEW tcp dpt:22
5 REJECT all -- 0.0.0.0/0 0.0.0.0/0 reject-with icmp-host-prohibited
6 ACCEPT udp -- 0.0.0.0/0 0.0.0.0/0 state NEW udp dpt:161

Chain FORWARD (policy ACCEPT)
num target prot opt source destination
1 REJECT all -- 0.0.0.0/0 0.0.0.0/0 reject-with icmp-host-prohibited

Chain OUTPUT (policy ACCEPT)
num target prot opt source destination
```

DO NOT FORGET to make these changes persistent by running **iptables save** command:

```
[root@geekpeek1 ~]# /etc/init.d/iptables save
iptables: Saving firewall rules to /etc/sysconfig/iptables:[ OK ]
```

Please note that here is **no need to reload iptables configuration when removing IPTables rules**!

4. Adding IPTables Rules (persistent)

By now you know how to configure iptables on your own. There is one more thing i would like to talk about.

I still think the best and easiest way to manage IPTables rules is to **manage it via iptables configuration file** located at **/etc/sysconfig/iptables**. The default fresh install /etc/sysconfig/iptables file is as follows:

```
[root@geekpeek1 ~]# cat /etc/sysconfig/iptables
 # Firewall configuration written by system-config-firewall
 # Manual customization of this file is not recommended.
 *filter
 :INPUT ACCEPT [0:0]
 :FORWARD ACCEPT [0:0]
 :OUTPUT ACCEPT [0:0]
 -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
 -A INPUT -p icmp -j ACCEPT
 -A INPUT -i lo -j ACCEPT
 -A INPUT -m state --state NEW -m tcp -p tcp --dport 22 -j ACCEPT
 -A INPUT -j REJECT --reject-with icmp-host-prohibited
 -A FORWARD -j REJECT --reject-with icmp-host-prohibited
 COMMIT
```

We manage IPTables configuration by **editing /etc/sysconfig/iptables** file and **reloading or restarting iptables service**.

Using your favorite linux console editor you can add or delete a rule. As we see there is already a defined **rule to accept SSH traffic** on port 22 which we can use to **copy and paste** and change the port nubmer to the desired one.

This way we can add all of the rules from *Step 2* which makes /etc/sysconfig/iptables file look like this:

```
[root@geekpeek1 ~]# cat /etc/sysconfig/iptables
 # Firewall configuration written by system-config-firewall
 # Manual customization of this file is not recommended.
 *filter
 :INPUT ACCEPT [0:0]
 :FORWARD ACCEPT [0:0]
 :OUTPUT ACCEPT [0:0]
 -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
 -A INPUT -p icmp -j ACCEPT
 -A INPUT -i lo -j ACCEPT
 -A INPUT -m state --state NEW -m tcp -p tcp --dport 22 -j ACCEPT
 -A INPUT -m state --state NEW -m tcp -p tcp --dport 80 -j ACCEPT
 -A INPUT -m state --state NEW -m udp -p udp --dport 161 -j ACCEPT
 -A INPUT -s 8.8.8.8 -m state --state NEW -m tcp -p tcp --dport 443 -j ACCEPT
 -A INPUT -j REJECT --reject-with icmp-host-prohibited
 -A FORWARD -j REJECT --reject-with icmp-host-prohibited
 COMMIT
```

..as said we **NEED** to **RELOAD** or **RESTART** the **iptables service**:

```
[root@geekpeek1 ~]# /etc/init.d/iptables reload
 iptables: Trying to reload firewall rules: [ OK ]
```

or

```
[root@geekpeek1 ~]# /etc/init.d/iptables restart
 iptables: Setting chains to policy ACCEPT: filter [ OK ]
 iptables: Flushing firewall rules: [ OK ]
 iptables: Unloading modules: [ OK ]
 iptables: Applying firewall rules: [ OK ]
```

..and check the **iptables status** to confirm the rules were applied:

```
[root@geekpeek1 ~]# /etc/init.d/iptables status
 Table: filter
 Chain INPUT (policy ACCEPT)
 num target prot opt source destination
 1 ACCEPT all -- 0.0.0.0/0 0.0.0.0/0 state RELATED,ESTABLISHED
 2 ACCEPT icmp -- 0.0.0.0/0 0.0.0.0/0
 3 ACCEPT all -- 0.0.0.0/0 0.0.0.0/0
 4 ACCEPT tcp -- 0.0.0.0/0 0.0.0.0/0 state NEW tcp dpt:22
 5 ACCEPT tcp -- 0.0.0.0/0 0.0.0.0/0 state NEW tcp dpt:80
 6 ACCEPT udp -- 0.0.0.0/0 0.0.0.0/0 state NEW udp dpt:161
 7 ACCEPT tcp -- 8.8.8.8 0.0.0.0/0 state NEW tcp dpt:443
 8 REJECT all -- 0.0.0.0/0 0.0.0.0/0 reject-with icmp-host-prohibited

Chain FORWARD (policy ACCEPT)
 num target prot opt source destination
 1 REJECT all -- 0.0.0.0/0 0.0.0.0/0 reject-with icmp-host-prohibited
```

```
1 REJECT all -- 0.0.0.0/0 0.0.0.0/0 reject-with icmp-host-prohibited

Chain OUTPUT (policy ACCEPT)
 num target prot opt source destination
```

5. Removing IPTables Rules (persistent)

This is our last step in the post on "*Configure IPTables on CentOS 6*". As said in *Step 4* we manage IPTables configuration by **editing /etc/sysconfig/iptables file** and after that reloading or restarting iptables service. This goes both for **adding new rules and also for removing rules**.

So this is what the /etc/sysconfig/iptables file looks like right now:

```
[root@geekpeek1 ~]# cat /etc/sysconfig/iptables
 # Firewall configuration written by system-config-firewall
 # Manual customization of this file is not recommended.
 *filter
 :INPUT ACCEPT [0:0]
 :FORWARD ACCEPT [0:0]
 :OUTPUT ACCEPT [0:0]
 -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
 -A INPUT -p icmp -j ACCEPT
 -A INPUT -i lo -j ACCEPT
 -A INPUT -m state --state NEW -m tcp -p tcp --dport 22 -j ACCEPT
 -A INPUT -m state --state NEW -m tcp -p tcp --dport 80 -j ACCEPT
 -A INPUT -m state --state NEW -m udp -p udp --dport 161 -j ACCEPT
```

```
-A INPUT -s 8.8.8.8 -m state --state NEW -m tcp -p tcp --dport 443 -j ACCEPT
-A INPUT -j REJECT --reject-with icmp-host-prohibited
-A FORWARD -j REJECT --reject-with icmp-host-prohibited
COMMIT
```

Use your favorite **linux console editor** to open /etc/sysconfig/iptables and remove the rules we added in *Step 4*:

```
[root@geekpeek1 ~]# cat /etc/sysconfig/iptables
# Firewall configuration written by system-config-firewall
# Manual customization of this file is not recommended.
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
-A INPUT -p icmp -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 22 -j ACCEPT
-A INPUT -j REJECT --reject-with icmp-host-prohibited
-A FORWARD -j REJECT --reject-with icmp-host-prohibited
COMMIT
```

RELOAD or **RESTART** the **iptables service**:

```
[root@geekpeek1 ~]# /etc/init.d/iptables reload
```

```
 iptables: Trying to reload firewall rules: [ OK ]
```

or

```
[root@geekpeek1 ~]# /etc/init.d/iptables restart
 iptables: Setting chains to policy ACCEPT: filter [ OK ]
 iptables: Flushing firewall rules: [ OK ]
 iptables: Unloading modules: [ OK ]
 iptables: Applying firewall rules: [ OK ]
```

..and check the **iptables status** to confirm the rules were applied:

```
[root@geekpeek1 ~]# /etc/init.d/iptables status
 Table: filter
 Chain INPUT (policy ACCEPT)
 num target prot opt source destination
 1 ACCEPT all -- 0.0.0.0/0 0.0.0.0/0 state RELATED,ESTABLISHED
 2 ACCEPT icmp -- 0.0.0.0/0 0.0.0.0/0
 3 ACCEPT all -- 0.0.0.0/0 0.0.0.0/0
 4 ACCEPT tcp -- 0.0.0.0/0 0.0.0.0/0 state NEW tcp dpt:22
 5 REJECT all -- 0.0.0.0/0 0.0.0.0/0 reject-with icmp-host-prohibited

Chain FORWARD (policy ACCEPT)
 num target prot opt source destination
 1 REJECT all -- 0.0.0.0/0 0.0.0.0/0 reject-with icmp-host-prohibited
```

```
Chain OUTPUT (policy ACCEPT)
 num target prot opt source destination
```

This is it for now! I hope you learned how to configure iptables. As said **there is much more to IPTables that this basic usage** and I will write another post where i will present you with the **IPTables security configuration you should use** on a server opened out to the internet! Stay tuned!

Tagged with: centos6 configure iptables security

- *gt00x-1a*
  Great intro … I finally understand ☺

- Pingback: Secure Your Linux Server With IPTables - GeekPeek.Net()

- Pingback: Install and Configure Postfix with Dovecot on CentOS 6 - GeekPeek.Net()

- *Sam Goodman*
  I must admit, this is the first IPtables post that makes 100% sense to me and has all the information I need!!!! Thanks so much!! Im sure there would be a million more thank you's if it wasnt so hard to post a comment. I had to register to post this, and thats something I never do.

  - *Mitch*
    Thanks Sam!

- *Saul Uribe*
  Great post, I already know about #iptables, but I can share it because is good and understandable. Can I translate to spanish? Thanks anyway

- *Mitch*

  Hi Saul and thanks. Sure you can translate to spanish if you like.