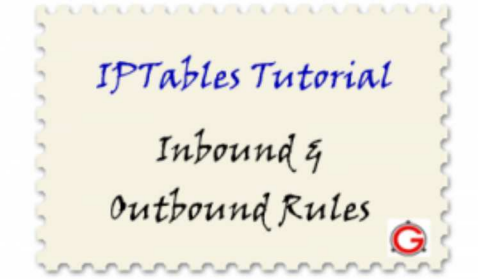
Linux IPTables: Incoming and Outgoing Rule Examples (SSH and HTTP)

by Ramesh Natarajan on March 15, 2011



In our previous IPTables firewall series article, we reviewed how to add firewall rule using "iptables -A".

We also explained how to allow incoming SSH connection. On a high-level, it involves following 3 steps.

1. Delete all existing rules: "iptables -F"
2. Allow only incoming SSH: "iptables -A INPUT -i eth0 -p tcp –dport 22 -j ACCEPT"
3. Drop all other incoming packets: "iptables -A INPUT -j DROP"

The above works. But it is not complete. One problem with the above steps is that it doesn't restrict the outgoing packets.

Default Chain Policy

The default policy of a chain is ACCEPT. If you don't what what a chain means, you better read our iptables introduction article. So, both the INPUT and OUTPUT chain's default policy is ACCEPT. In the above 3 steps we dropped all incoming packets at the end (except incoming ssh). However, we didn't restrict the outgoing traffic.

As you notice below, it says "(policy ACCEPT)" next to all the three chain names (INPUT, OUTPUT, and FORWARD). This indicates that the default chain policy is ACCEPT.

```
# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     tcp  --  anywhere             anywhere            tcp dpt:ssh
DROP       all  --  anywhere             anywhere

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

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

So, you have two options here.

Option 1: Add drop rules

At the end, add the following three drop rules that will drop all incoming, outgoing, and forward packets (except those that are defined above these three rules). If you do this, the default chain policy is still ACCEPT, which shouldn't matter, as you are dropping all the packets at the end anyway.

```
iptables -A INPUT -j DROP
iptables -A OUTPUT -j DROP
iptables -A FORWARD -j DROP
```

Option 2: Change the default chain policy to DROP

At the beginning, execute the following three commands that will change the chain's default policy to DROP.

```
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP
```

Now, if you add the allow ssh rule: "iptables -A INPUT -i eth0 -p tcp –dport 22 -j ACCEPT", and do iptables -L, you'll notice that it says "(policy DROP)" next to all the three chains.

```
# iptables -L
Chain INPUT (policy DROP)
target     prot opt source               destination
ACCEPT     tcp  --  anywhere             anywhere            tcp dpt:ssh
DROP       all  --  anywhere             anywhere

Chain FORWARD (policy DROP)
target     prot opt source               destination

Chain OUTPUT (policy DROP)
target     prot opt source               destination
```

But there is a problem here. The allow ssh incoming connection rule will not work anymore, because all the outgoing packets are dropped.

Allow Incoming Connections

When the default policy is DROP for INPUT and OUTPUT chains, for every incoming firewall rule, you need to specify the following two rules.

1. Request rule: This is the request that comes from the client to the server for the incoming connection.
2. Response rule: This is for the response that goes out from the server to the client (for the corresponding incoming request).

Example 1: Allow incoming SSH connection

This is to allow SSH connection from outside to your server. i.e You can ssh to your server from outside.

This involves two steps. First, we need to allow incoming new SSH connections. Once the incoming ssh connection is allowed, we also need to allow the response back for that incoming ssh connection.

First, Allow incoming SSH connection request, as shown below.

```
iptables -A INPUT -i eth0 -p tcp --dport 22 -m state --state NEW,ESTABLISHED -j ACCEPT
```

In the above example:

- iptables -A INPUT: Append the new rule to the INPUT chain. For incoming connection request, this always has to be INPUT.
- -i eth0: This refers to the input interface. For incoming connections, this always has to be '-i'.
- -p tcp: Indicates that this is for TCP protocol.
- –dport 22: This refers to the destination port for the incoming connection. Port 22 is for ssh.
- -m state: This indicates that the "state" matching module is used. We'll discuss more about "-m" option (and all available matching modules for iptables) in future article.
- –state NEW, ESTABLISHED: Options for the "state" matching module. In this example, only NEW and ESTABLISHED states are allowed. The 1st time when a SSH connection request is initiated from the client to the server, NEW state is used. ESTABLISHED state is used for all further request from the client to the server.

Next, Allow outgoing (ESTABLISHED state only) SSH connection response (for the corresponding incoming SSH connection request).

```
iptables -A OUTPUT -o eth0 -p tcp --sport 22 -m state --state ESTABLISHED -j ACCEPT
```

In the above example:

- iptables -A OUTPUT: Append the new rule to the OUTPUT chain. Since this is for the response rule (for the corresponding incoming request) that goes out from the server, this should be OUTPUT.
- -o eth0: This refers the output interface. For outgoing connections, this always has to be '-o'.
- -p tcp: Indicates that this is for TCP protocol.
- –sport 22: This refers to the source port for the outgoing connection. Port 22 is for ssh. Since the incoming request (from the previous rule) came to the "destination" port, the outgoing response will go through the "source" port.
- -m state: This indicates that the "state" matching module is used.
- –state ESTABLISHED: Since this is a response rule, we allow only ESTABLISHED connection (and not any NEW connection).

Example 2: Allow incoming HTTP connection

This is to allow HTTP connection from outside to your server. i.e You can view your website running on the server from outside.

Just like the above SSH incoming rules, this also involves two steps. First, we need to allow incoming new HTTP connection. Once the incoming HTTP connection is allowed, we need to allow the response back for that incoming HTTP connection.

First, Allow incoming HTTP connection request, as shown below.

```
iptables -A INPUT -i eth0 -p tcp --dport 80 -m state --state NEW,ESTABLISHED -j ACCEPT
```

Next, Allow outgoing (ESTABLISHED only) HTTP connection response (for the corrresponding incoming SSH connection request).

```
iptables -A OUTPUT -o eth0 -p tcp --sport 80 -m state --state ESTABLISHED -j ACCEPT
```

Note: In the above HTTP request and response rule, everything is same as the SSH example except the port number.

Allow Outgoing Connections

When the default policy is DROP for the INPUT and OUTPUT chains, for every outgoing firewall rule, you need to specify the following two rules.

1. Request rule: This is the request that goes out from the server to outside for the outgoing connection.
2. Response rule: This is for the response that comes back from the outside to the server (for the corresponding outgoing request).

Example 3: Allow outgoing SSH connection

This is to allow SSH connection from your server to the outside. i.e You can ssh to outside server from your server.

This involves two steps. First, we need to allow outgoing new SSH connection. Once the outgoing ssh connection is allowed, we also need to allow the response back for that outgoing ssh connection.

First, Allow outgoing SSH connection request, as shown below.

```
iptables -A OUTPUT -o eth0 -p tcp --dport 22 -m state --state NEW,ESTABLISHED -j ACCEPT
```

In the above example:

- iptables -A OUTPUT: Append the new rule to the OUTPUT chain. For outgoing connection request, this always has to be OUTPUT.
- -o eth0: This refers the output interface. For outgoing connections, this always has to be '-o'.
- -p tcp: Indicates that this is for TCP protocol.
- –dport 22: This refers to the destination port for the outgoing connection.
- -m state: This indicates that "state" matching module is used.
- –state NEW, ESTABLISHED: Options for the "state" matching module. In this example, only NEW and ESTABLISHED states are allowed. The 1st time when a SSH connection request is initiated from the server to the outside, NEW state is used. ESTABLISHED state is used for all further request from the server to the outside.

Next, Allow outgoing (ESTABLISHED only) SSH connection response (for the corresponding incoming SSH connection request).

```
iptables -A INPUT -i eth0 -p tcp --sport 22 -m state --state ESTABLISHED -j ACCEPT
```

In the above example:

- iptables -A INPUT: Append the new rule to the INPUT chain. Since this is for the response rule (for the corresponding outgoing request) that comes from the outside to the server, this should be INPUT.
- -i eth0: This refers the input interface. For incoming connections, this always has to be '-i'.
- -p tcp: Indicates that this is for TCP protocol.
- –sport 22: This refers to the source port for the incoming connection. Since the outgoing request (from the previous rule) went to the "destination" port, the incoming response will come from the "source" port.
- -m state: This indicates that the "state" matching module is used.
- –state ESTABLISHED: Since this is a response rule, we allow only ESTABLISHED connection (and not any NEW connection).

Putting it all together

Create rules.sh shell script which does the following:

1. Delete all existing rules
2. Set default chain policies
3. Allow inbound SSH
4. Allow inbound HTTP
5. Allow outbound SSH

First, create the rules.sh

```
$ vi rules.sh
# 1. Delete all existing rules
iptables -F

# 2. Set default chain policies
iptables -P INPUT DROP
iptables -P FORWARD DROP
iptables -P OUTPUT DROP

# 3. Allow incoming SSH
iptables -A INPUT -i eth0 -p tcp --dport 22 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -o eth0 -p tcp --sport 22 -m state --state ESTABLISHED -j ACCEPT

# 4. Allow incoming HTTP
iptables -A INPUT -i eth0 -p tcp --dport 80 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -o eth0 -p tcp --sport 80 -m state --state ESTABLISHED -j ACCEPT

# 5. Allow outgoing SSH
iptables -A OUTPUT -o eth0 -p tcp --dport 22 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A INPUT -i eth0 -p tcp --sport 22 -m state --state ESTABLISHED -j ACCEPT
```

Next, execute the rules.sh and view the rules.

```
# chmod u+x rules.sh

# ./rules.sh

# iptables -L
Chain INPUT (policy DROP)
target     prot opt source      destination
ACCEPT     tcp  --  anywhere    anywhere      tcp dpt:ssh state NEW,ESTABLISHED
ACCEPT     tcp  --  anywhere    anywhere      tcp dpt:http state NEW,ESTABLISHED
ACCEPT     tcp  --  anywhere    anywhere      tcp spt:ssh state ESTABLISHED

Chain FORWARD (policy DROP)
target     prot opt source      destination

Chain OUTPUT (policy DROP)
target     prot opt source      destination
ACCEPT     tcp  --  anywhere    anywhere      tcp spt:ssh state ESTABLISHED
ACCEPT     tcp  --  anywhere    anywhere      tcp spt:http state ESTABLISHED
ACCEPT     tcp  --  anywhere    anywhere      tcp dpt:ssh state NEW,ESTABLISHED
```

Using this as a basis you should be able to write your own incoming and outgoing iptables firewall rules. There is lot more to cover in IPTables. Stay tuned!

Previous articles in the iptables series:

- Linux Firewall Tutorial: IPTables Tables, Chains, Rules Fundamentals
- IPTables Flush: Delete / Remove All Rules On RedHat and CentOS Linux
- Linux IPTables: How to Add Firewall Rules (With Allow SSH Example)

If you enjoyed this article, you might also like..

1. 50 Linux Sysadmin Tutorials
2. 50 Most Frequently Used Linux Commands (With Examples)
3. Top 25 Best Linux Performance Monitoring and Debugging Tools
4. Mommy, I found it! – 15 Practical Linux Find Command Examples
5. Linux 101 Hacks 2nd Edition eBook Free

- Awk Introduction – 7 Awk Print Examples
- Advanced Sed Substitution Examples
- 8 Essential Vim Editor Navigation Fundamentals
- 25 Most Frequently Used Linux IPTables Rules Examples
- Turbocharge PuTTY with 12 Powerful Add-Ons