# iptable Introduction

## iptables Simplified

This introduction is intended for people trying to get their system configured without spending a lot of time learning the intricacies of iptables. iptables define a set of rules that tell the kernel how to handle network traffic; what things should be allowed to go in or out of the system and for more advanced uses, what changes should be made to a packet as it passes through (NAT filtering). This means that you can specify that traffic for certain ports should be allowed or denied, or that traffic from or to certain hosts should be allowed or denied.
For example, you could allow no telnet traffic, or you could allow telnet traffic only from hosts in your local domain. You could deny any attempt by a user to connect using ssh to your mail server, but you could still allow the mail traffic to follow the same path. This is a very powerful capability when you need to protect your system and users.
The first thing you need to do is find out the commands and file system locations of the pertinent information.

- *iptables* is the command that you use to make dynamic changes to your iptables.
- *iptables-save* saves the current iptables to a file.
- *iptables-restore* restores the current iptables from a file.
- */etc/sysconfig/iptables* is the normal storage location for the system's iptables configuration.
- *ipfwadm* is a common tool for performing iptable configuration.

If you want to see what your current iptables are, enter the following command:

```
iptables -L
```

What you see is something like this:

```
Chain INPUT (policy ACCEPT)
target     prot opt source                    destination
ACCEPT     tcp  --  153.90.192.0/21   anywhere     tcp dpt:ipp flags:SYN,RST,ACK/SYN
ACCEPT     tcp  --  153.90.192.0/21   anywhere     tcp dpt:http flags:SYN,RST,ACK/SYN
ACCEPT     tcp  --  153.90.192.0/21   anywhere     tcp dpt:ssh flags:SYN,RST,ACK/SYN
ACCEPT     tcp  --  153.90.192.0/21   anywhere     tcp dpt:netbios-ns flags:SYN,RST,ACK/SY
ACCEPT     tcp  --  153.90.192.0/21   anywhere     tcp dpt:netbios-ssn flags:SYN,RST,ACK/S
ACCEPT     tcp  --  153.90.192.0/21   anywhere     state RELATED,ESTABLISHED
ACCEPT     udp  --  153.90.192.0/21   anywhere     state RELATED,ESTABLISHED
ACCEPT     tcp  --  153.90.192.0/21   anywhere     tcp dpt:nfs
ACCEPT     udp  --  153.90.192.0/21   anywhere     udp dpt:nfs
ACCEPT     udp  --  153.90.192.0/21   anywhere     udp dpt:sunrpc
           all  --  anywhere          anywhere
RH-Lokkit-0-50-INPUT  all  --  anywhere             anywhere

Chain FORWARD (policy ACCEPT)
target     prot opt source            destination
RH-Lokkit-0-50-INPUT  all  --  anywhere             anywhere

Chain OUTPUT (policy ACCEPT)
target     prot opt source            destination

Chain RH-Lokkit-0-50-INPUT (2 references)
target     prot opt source            destination
ACCEPT     udp  --  dns1.msu.montana.edu  anywhere     udp spt:domain dpts:1025:65535
ACCEPT     udp  --  coesrv.coe.montana.edu  anywhere   udp spt:domain dpts:1025:65535
REJECT     tcp  --  anywhere              anywhere     tcp dpts:0:1023 flags:SYN,RST,ACK/S
REJECT     udp  --  anywhere              anywhere     udp dpts:0:1023 reject-with icmp-po
REJECT     tcp  --  anywhere              anywhere     tcp dpts:x11:6009 flags:SYN,RST,ACK
REJECT     tcp  --  anywhere              anywhere     tcp dpt:xfs flags:SYN,RST,ACK/SYN r
```

## iptables Rules

The file is divided into four chains, each of which has a set of rules. In this case the chains are: *INPUT, FORWARD, OUTPUT, RH-Lokkit-0-50-INPUT*. The first three are standard iptables chains while the last is a user defined chain that is referenced as the last rule in the INPUT chain. So what you see under the last chain are actually INPUT chain rules. For an introduction, all you need to worry about is the input chain (thankfully).
The INPUT chain contains rules that define what network traffic will be allowed or denied to enter the system. Each rule has the following structure:

| | |
|---|---|
| ACCEPT | any packet meeting these specs will be allowed in. Other possibile actions are REJECT and DROP. |
| protocol | what protocols; e.g. tcp, udp, all |
| options | could be a variety of things |
| source IP | the IP address of the sender |
| destination IP | the IP address of the receiver |
| destination | the protocol, port number and state flags |

Some of this seems a little strange. For example, the destination IP address is always the current system, right? No, if you have a system acting as a router the destination could be some other system.
To understand what this means, look at the second rule. It says that any input packet that comes from the local network and is destined for port *http* (the web server) should be accepted. It is actually a little more complicated than that because the flags specify that only TCP packets that are *SYN, RST* or *ACK/SYN* are to be accepted. These are packets that are used to open a connection. The other packets are accepted by a later rule that accepts any packet that has a state of *RELATED, ESTABLISHED*. So once a connection is allowed to be open, following packets for that connection will be allowed.

## Changing Your iptables

So how do you delete or add rules? You should remember that the iptables are a dynamic part of the kernel, so when you make a change, it is immediately inserted into the tables being used. Rules are added like this, where packets for sshd are allowed:

```
iptables -A INPUT -p tcp –dport 22 –syn -j ACCEPT
```

This produces the third rule in the list above.
Of some importance is your understanding of how these rules are processed. The order you see when you type iptables -L is the order in which the rules are processed. If a rule matches, the action is taken, so if the rule says ACCEPT, REJECT, DROP or something else, that happens and the packet processing is finished. If it doesn't match, the next rule is applied and so on. If you want to get rid of everything, you can use a rule like this:

```
iptables -A INPUT -s anywhere -d anywhere -j REJECT
```

Once a packet is matched by any rule it is accepted, or denied and no further processing is done, so the rules should be organized so that packets are denied first and if they survive, they might be accepted. Also, ACCEPT's should be order from the specific to the general.
An example of some rules and their meaning are:

iptables -I INPUT -p tcp -s 153.90.199.0/21 -d any –dport ssh -j ACCEPT
: allow ssh requests from hosts in the local network, as defined by the first 21 bits of the IP addre
iptables -I INPUT -p tcp -s 192.168.2.5 -d any –dport ssh -j ACCEPT
: allow ssh requests from 192.168.2.5. This allows requests from a host that does not meet the fi
iptables -I INPUT -p tcp -s any -d any –dport 1450:1500 -j ACCEPT
: allow packets to ports 1450 to 1500 from any host. This is normally not a good idea.
iptables -I INPUT -p any -s 192.168.2.5 -d any –dport 109:110 -j DENY
: don't allow packets from this host for the POP services.

After setting up your iptables the way you want them, you should save them so that they can be restored when you reboot. Remember that this doesn't happen automagically. To do this,

iptables-save ¿ /etc/sysconfig/iptables

You can save the tables anywhere, but the path shown above is the default and that is where they are loaded from during booting.
Another way to approach iptables changes is to edit the iptables file and then restore the tables. You can create the rules that you like, following the format in the file, and then activate the changes with. For example, the rule below in the iptables file allows limited access to the sshd port:

-A INPUT -p tcp -m tcp -s 153.90.192.0/21 –dport 22 –syn -j ACCEPT

and the rule is made current with:

iptables-restore ¡ /etc/sysconfig/iptables

## Some Interesting Rules

One of the things that you can do in the iptables is log all requests so that you can identify potential attackers. The following rules log all requests.
Logging requires that you specify the qualities of the packets to be logged and their log level. For example,

```
# Turn on logging for foreign addresses
# -A INPUT -p tcp -s ! 153.90.192.0/21 -j LOG –log-level warning
# -A INPUT -p udp -s ! 153.90.192.0/21 -j LOG
```

iptables logging is a kernel message, so you can look in /etc/syslog.conf to see where the messages go. The default is probably to /var/log/messages which is not a great place, but OK temporarily.
You might also want to allow all traffic from the local system.

```
# Accept all local traffic
-A INPUT -i lo -j ACCEPT
```

Or deny all traffic that hasn't been accepted.

```
# Deny everything that gets here
-A INPUT -s 0.0.0.0 -d 0.0.0.0 -j REJECT
```

## A Basic Set of Rules

For a system just getting started, here is a basic iptables file that works well, allowing traffic only from the localhost and for ssh from the local domain.

```
# Turn on logging for foreign addresses
# -A INPUT -p tcp -s ! 153.90.192.0/21 -j LOG
# -A INPUT -p udp -s ! 153.90.192.0/21 -j LOG

# Accept all local traffic and ssh
-A INPUT -i lo -j ACCEPT
-A INPUT -p tcp -m tcp -s 153.90.192.0/21 --dport 22 --syn -j ACCEPT

# Make sure certain types of things are rejected that haven't been
# already accepted.
-A INPUT -p tcp -m tcp --dport 0:1023 --syn -j REJECT
-A INPUT -p udp -m udp --dport 0:1023 -j REJECT
-A INPUT -p tcp -m tcp --dport 6000:6009 --syn -j REJECT

# Accept all forward and output traffic
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
```