Table Of Contents

nixCraft: Linux Tips, Hacks, Tutorials, And Ideas In Blog Format
http://www.cyberciti.biz/

CentOS / Redhat Iptables Firewall Configuration Tutorial

Posted by Vivek Gite <vivek@nixcraft.com>


[1]

How do I configure a host-based firewall called Netfilter (iptables) under CentOS / RHEL / Fedora / Redhat Enterprise Linux?

Netfilter is a host-based firewall for Linux operating systems. It is included as part of the Linux distribution and it is activated by default. This firewall is controlled by the program called iptables. Netfilter filtering take place at the kernel level, before a program can even process the data from the network packet.

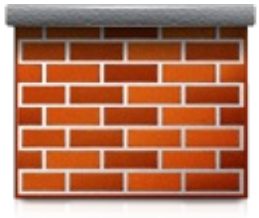
Iptables Config File


[2]

The default config files for RHEL / CentOS / Fedora Linux are:

- /etc/sysconfig/iptables - The system scripts that activate the firewall by reading this file.

Task: Display Default Rules


[3]

Type the following command:

```
iptables --line-numbers -n -L
```

Sample outputs:

```
Chain INPUT (policy ACCEPT)
num  target     prot opt source               destination
1    RH-Firewall-1-INPUT  all  --  0.0.0.0/0            0.0.0.0/0

Chain FORWARD (policy ACCEPT)
num  target     prot opt source               destination
1    RH-Firewall-1-INPUT  all  --  0.0.0.0/0            0.0.0.0/0

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

Chain RH-Firewall-1-INPUT (2 references)
num  target     prot opt source               destination
1    ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0
2    ACCEPT     icmp --  0.0.0.0/0            0.0.0.0/0           icmp type 255
3    ACCEPT     udp  --  0.0.0.0/0            224.0.0.251         udp dpt:5353
4    ACCEPT     udp  --  0.0.0.0/0            0.0.0.0/0           udp dpt:53
5    ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0           state RELATED,ESTABLISHED
6    ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0           state NEW tcp dpt:22
7    ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0           state NEW tcp dpt:53
8    REJECT     all  --  0.0.0.0/0            0.0.0.0/0           reject-with icmp-host-prohibited
```

Task: Turn On Firewall

Type the following two commands to turn on firewall:

```
chkconfig iptables on
```

```
service iptables start
# restart the firewall
service iptables restart
# stop the firewall
service iptables stop
```

Understanding Firewall

There are total 4 chains:

1. **INPUT** - The default chain is used for packets addressed to the system. Use this to open or close incoming ports (such as 80,25, and 110 etc) and ip addresses / subnet (such as 202.54.1.20/29).
2. **OUTPUT** - The default chain is used when packets are generating from the system. Use this open or close outgoing ports and ip addresses / subnets.
3. **FORWARD** - The default chains is used when packets send through another interface. Usually used when you setup Linux as router. For example, eth0 connected to ADSL/Cable modem and eth1 is connected to local LAN. Use FORWARD chain to send and receive traffic from LAN to the Internet.
4. **RH-Firewall-1-INPUT** - This is a user-defined custom chain. It is used by the INPUT, OUTPUT and FORWARD chains.

Packet Matching Rules

1. Each packet starts at the first rule in the chain .
2. A packet proceeds until it matches a rule.
3. If a match found, then control will jump to the specified target (such as REJECT, ACCEPT, DROP).

Target Meanings

1. The target **ACCEPT** means allow packet.
2. The target **REJECT** means to drop the packet and send an error message to remote host.
3. The target **DROP** means drop the packet and do not send an error message to remote host or sending host.

/etc/sysconfig/iptables

Edit /etc/sysconfig/iptables, enter:

```
# vi /etc/sysconfig/iptables
```

You will see default rules as follows:

```
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
:RH-Firewall-1-INPUT - [0:0]
-A INPUT -j RH-Firewall-1-INPUT
-A FORWARD -j RH-Firewall-1-INPUT
-A RH-Firewall-1-INPUT -i lo -j ACCEPT
-A RH-Firewall-1-INPUT -p icmp --icmp-type any -j ACCEPT
-A RH-Firewall-1-INPUT -p udp --dport 5353 -d 224.0.0.251 -j ACCEPT
-A RH-Firewall-1-INPUT -p udp -m udp --dport 53 -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 22 -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 53 -j ACCEPT
-A RH-Firewall-1-INPUT -j REJECT --reject-with icmp-host-prohibited
COMMIT
```

Drop All Traffic

Find lines:

```
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
```

Update as follows to change the default policy to DROP from ACCEPT for the INPUT and FORWARD built-in chains:

```
:INPUT DROP [0:0]
:FORWARD DROP [0:0]
```

Log and Drop Spoofing Source Addresses

Append the following lines before final COMMIT line:

```
-A INPUT -i eth0 -s 10.0.0.0/8 -j LOG --log-prefix "IP DROP SPOOF "
-A INPUT -i eth0 -s 172.16.0.0/12 -j LOG --log-prefix "IP DROP SPOOF "
-A INPUT -i eth0 -s 192.168.0.0/16 -j LOG --log-prefix "IP DROP SPOOF "
-A INPUT -i eth0 -s 224.0.0.0/4 -j LOG --log-prefix "IP DROP MULTICAST "
-A INPUT -i eth0 -s 240.0.0.0/5 -j LOG --log-prefix "IP DROP SPOOF "
-A INPUT -i eth0 -d 127.0.0.0/8 -j LOG --log-prefix "IP DROP LOOPBACK "
-A INPUT -i eth0 -s 169.254.0.0/16  -j LOG --log-prefix "IP DROP MULTICAST "
-A INPUT -i eth0 -s 0.0.0.0/8  -j LOG --log-prefix "IP DROP "
-A INPUT -i eth0 -s  240.0.0.0/4  -j LOG --log-prefix "IP DROP "
-A INPUT -i eth0 -s  255.255.255.255/32  -j LOG --log-prefix "IP DROP  "
-A INPUT -i eth0 -s 168.254.0.0/16  -j LOG --log-prefix "IP DROP "
-A INPUT -i eth0 -s 248.0.0.0/5  -j LOG --log-prefix "IP DROP "
```

Log And Drop All Traffic

Find the lines:

```
-A RH-Firewall-1-INPUT -j REJECT --reject-with icmp-host-prohibited
COMMIT
```

Update it as follows:

```
-A RH-Firewall-1-INPUT -j LOG
-A RH-Firewall-1-INPUT -j DROP
COMMIT
```

Open Port

To open port 80 (Http server) add the following before COMMIT line:

```
-A RH-Firewall-1-INPUT -m tcp -p tcp --dport 80 -j ACCEPT
```

To open port 53 (DNS Server) add the following before COMMIT line:

```
-A RH-Firewall-1-INPUT -m tcp -p tcp --dport 53 -j ACCEPT
-A RH-Firewall-1-INPUT -m udp -p tcp --dport 53 -j ACCEPT
```

To open port 443 (Https server) add the following before COMMIT line:

```
-A RH-Firewall-1-INPUT -m tcp -p tcp --dport 443 -j ACCEPT
```

To open port 25 (smtp server) add the following before COMMIT line:

```
-A RH-Firewall-1-INPUT -m tcp -p tcp --dport 25 -j ACCEPT
```

Only allow SSH traffic From 192.168.1.0/24

```
-A RH-Firewall-1-INPUT -s 192.168.1.0/24 -m state --state NEW -p tcp --dport 22 -j ACCEPT
```

Enable Printing Access For 192.168.1.0/24

```
-A RH-Firewall-1-INPUT -s 192.168.1.0/24 -p udp -m udp --dport 631 -j ACCEPT
-A RH-Firewall-1-INPUT -s 192.168.1.0/24 -p tcp -m tcp --dport 631 -j ACCEPT
```

Allow Legitimate NTP Clients to Access the Server

```
-A RH-Firewall-1-INPUT -s 192.168.1.0/24 -m state --state NEW -p udp --dport 123 -j ACCEPT
```

Open FTP Port 21 (FTP)

```
-A RH-Firewall-1-INPUT -m state --state NEW -p tcp --dport 21 -j ACCEPT
```

Save and close the file. Edit /etc/sysconfig/iptables-config, enter:

```
# vi /etc/sysconfig/iptables-config
```

Make sure ftp module is loaded with the space-separated list of modules:

```
IPTABLES_MODULES="ip_conntrack_ftp"
```

To restart firewall, type the following commands:

```
# service iptables restart
# iptables -vnL --line-numbers
```

Edit /etc/sysctl.conf For DoS and Syn Protection

Edit /etc/sysctl.conf to defend against certain types of attacks and append / update as follows:

```
et.ipv4.conf.all.log_martians = 1
net.ipv4.conf.default.accept_source_route = 0
net.ipv4.conf.default.accept_redirects = 0
net.ipv4.conf.default.secure_redirects = 0
net.ipv4.icmp_echo_ignore_broadcasts = 1
#net.ipv4.icmp_ignore_bogus_error_messages = 1
net.ipv4.tcp_syncookies = 1
net.ipv4.conf.all.rp_filter = 1
net.ipv4.conf.default.rp_filter = 1
```

See previous FAQ, "Linux Kernel /etc/sysctl.conf [4] Security Hardening" for more details.

Alternate Configuration Option

You can skip /etc/sysconfig/iptables file and create a shell script from scratch as follows:

```
#!/bin/bash
# A sample firewall shell script
IPT="/sbin/iptables"
SPAMLIST="blockedip"
SPAMDROPMSG="BLOCKED IP DROP"
SYSCTL="/sbin/sysctl"
BLOCKEDIPS="/root/scripts/blocked.ips.txt"

# Stop certain attacks
echo "Setting sysctl IPv4 settings..."
$SYSCTL net.ipv4.ip_forward=0
$SYSCTL net.ipv4.conf.all.send_redirects=0
$SYSCTL net.ipv4.conf.default.send_redirects=0
```

```
$SYSCTL net.ipv4.conf.all.accept_source_route=0
$SYSCTL net.ipv4.conf.all.accept_redirects=0
$SYSCTL net.ipv4.conf.all.secure_redirects=0
$SYSCTL net.ipv4.conf.all.log_martians=1
$SYSCTL net.ipv4.conf.default.accept_source_route=0
$SYSCTL net.ipv4.conf.default.accept_redirects=0
$SYSCTL net.ipv4.conf.default.secure_redirects=0
$SYSCTL net.ipv4.icmp_echo_ignore_broadcasts=1
#$SYSCTL net.ipv4.icmp_ignore_bogus_error_messages=1
$SYSCTL net.ipv4.tcp_syncookies=1
$SYSCTL net.ipv4.conf.all.rp_filter=1
$SYSCTL net.ipv4.conf.default.rp_filter=1
$SYSCTL kernel.exec-shield=1
$SYSCTL kernel.randomize_va_space=1

echo "Starting IPv4 Firewall..."
$IPT -F
$IPT -X
$IPT -t nat -F
$IPT -t nat -X
$IPT -t mangle -F
$IPT -t mangle -X

# load modules
modprobe ip_conntrack

[ -f "$BLOCKEDIPS" ] && BADIPS=$(egrep -v -E "^#|^$" "${BLOCKEDIPS}")

# interface connected to the Internet
PUB_IF="eth0"

#Unlimited traffic for loopback
$IPT -A INPUT -i lo -j ACCEPT
$IPT -A OUTPUT -o lo -j ACCEPT

# DROP all incomming traffic
$IPT -P INPUT DROP
$IPT -P OUTPUT DROP
$IPT -P FORWARD DROP

if [ -f "${BLOCKEDIPS}" ];
then
# create a new iptables list
$IPT -N $SPAMLIST

for ipblock in $BADIPS
do
   $IPT -A $SPAMLIST -s $ipblock -j LOG --log-prefix "$SPAMDROPMSG "
   $IPT -A $SPAMLIST -s $ipblock -j DROP
done

$IPT -I INPUT -j $SPAMLIST
$IPT -I OUTPUT -j $SPAMLIST
$IPT -I FORWARD -j $SPAMLIST
fi

# Block sync
$IPT -A INPUT -i ${PUB_IF} -p tcp ! --syn -m state --state NEW  -m limit --limit 5/m --limi
$IPT -A INPUT -i ${PUB_IF} -p tcp ! --syn -m state --state NEW -j DROP

# Block Fragments
$IPT -A INPUT -i ${PUB_IF} -f  -m limit --limit 5/m --limit-burst 7 -j LOG --log-level 4 --
$IPT -A INPUT -i ${PUB_IF} -f -j DROP

# Block bad stuff
$IPT  -A INPUT -i ${PUB_IF} -p tcp --tcp-flags ALL FIN,URG,PSH -j DROP
$IPT  -A INPUT -i ${PUB_IF} -p tcp --tcp-flags ALL ALL -j DROP

$IPT  -A INPUT -i ${PUB_IF} -p tcp --tcp-flags ALL NONE -m limit --limit 5/m --limit-burst
$IPT  -A INPUT -i ${PUB_IF} -p tcp --tcp-flags ALL NONE -j DROP # NULL packets
```

```
$IPT  -A INPUT -i ${PUB_IF} -p tcp --tcp-flags SYN,RST SYN,RST -j DROP

$IPT  -A INPUT -i ${PUB_IF} -p tcp --tcp-flags SYN,FIN SYN,FIN -m limit --limit 5/m --limit
$IPT  -A INPUT -i ${PUB_IF} -p tcp --tcp-flags SYN,FIN SYN,FIN -j DROP #XMAS

$IPT  -A INPUT -i ${PUB_IF} -p tcp --tcp-flags FIN,ACK FIN -m limit --limit 5/m --limit-bur
$IPT  -A INPUT -i ${PUB_IF} -p tcp --tcp-flags FIN,ACK FIN -j DROP # FIN packet scans

$IPT  -A INPUT -i ${PUB_IF} -p tcp --tcp-flags ALL SYN,RST,ACK,FIN,URG -j DROP

# Allow full outgoing connection but no incomming stuff
$IPT -A INPUT -i ${PUB_IF} -m state --state ESTABLISHED,RELATED -j ACCEPT
$IPT -A OUTPUT -o ${PUB_IF} -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT

# Allow ssh
$IPT -A INPUT -i ${PUB_IF} -p tcp --destination-port 22 -j ACCEPT

# Allow http / https (open port 80 / 443)
$IPT -A INPUT -i ${PUB_IF} -p tcp --destination-port 80 -j ACCEPT
#$IPT -A INPUT -o ${PUB_IF} -p tcp --destination-port 443 -j ACCEPT

# allow incomming ICMP ping pong stuff
$IPT -A INPUT -i ${PUB_IF} -p icmp --icmp-type 8 -m state --state NEW,ESTABLISHED,RELATED
#$IPT -A OUTPUT -o ${PUB_IF} -p icmp --icmp-type 0 -m state --state ESTABLISHED,RELATED -j

# Allow port 53 tcp/udp (DNS Server)
$IPT -A INPUT -i ${PUB_IF} -p udp --dport 53 -m state --state NEW,ESTABLISHED,RELATED -j AC
#$IPT -A OUTPUT -o ${PUB_IF} -p udp --sport 53 -m state --state ESTABLISHED,RELATED -j ACC

$IPT -A INPUT -i ${PUB_IF} -p tcp --destination-port 53 -m state --state NEW,ESTABLISHED,RE
#$IPT -A OUTPUT -o ${PUB_IF} -p tcp --sport 53 -m state --state ESTABLISHED,RELATED -j ACC

# Open port 110 (pop3) / 143
$IPT -A INPUT -i ${PUB_IF} -p tcp --destination-port 110 -j ACCEPT
$IPT -A INPUT -i ${PUB_IF} -p tcp --destination-port 143 -j ACCEPT

##### Add your rules below ######
#
#
##### END your rules ############

# Do not log smb/windows sharing packets - too much logging
$IPT -A INPUT -p tcp -i ${PUB_IF} --dport 137:139 -j REJECT
$IPT -A INPUT -p udp -i ${PUB_IF} --dport 137:139 -j REJECT

# log everything else and drop
$IPT -A INPUT -j LOG
$IPT -A FORWARD -j LOG
$IPT -A INPUT -j DROP

exit 0
```

Recommend readings:

- See all our iptables related FAQs [3], tutorials [5], and shell scripts [6].
- Ip6tables (IPv6) firewall [7] configurations.
- Read iptables and sysctl man pages.

URLs in this post:

[1] Image: **http://www.cyberciti.biz/faq/category/centos/**
[2] Image: **http://www.cyberciti.biz/faq/category/redhat-and-friends/**
[3] Image: **http://www.cyberciti.biz/faq/category/iptables/**
[4] Linux Kernel /etc/sysctl.conf: **http://www.cyberciti.biz/faq/linux-kernel-etcsysctl-conf-security-hardening/**
[5] tutorials: **http://www.cyberciti.biz/tips/category/iptables**
[6] shell scripts: **http://bash.cyberciti.biz/shell/firewall/**
[7] Ip6tables (IPv6) firewall: **http://www.cyberciti.biz/faq/redhat-fedora-ip6tables-firewall-configuration/**
