iptables: a simple cheatsheet

May 8, 2019

Whether you're a novice user or a system administrator, iptables is a mandatory knowledge!

iptables is the userspace command line program used to configure the Linux 2.4.x and later packet filtering ruleset.
When a connection tries to establish itself on your system, iptables looks for a rule in its list to match it to.
If it doesn't find one, it resorts to the default action.

How it works

iptables uses three different chains to allow or block traffic: **input**, **output** and **forward**

- **Input** – This chain is used to control the behavior for incoming connections.
- **Output** – This chain is used for outgoing connections.
- **Forward** – This chain is used for incoming connections that aren't actually being delivered locally like routing and NATing.

Let's start to configure rules

By default all chains are configured to the accept rule, so during the hardening process the suggestion is to start with a deny all configuration and then open only needed ports:

```
iptables --policy INPUT DROP
iptables --policy OUTPUT DROP
iptables --policy FORWARD DROP
```

Display rules

Verbose print out **all active iptables rules**

```
# iptables -n -L -v
```

…same output with line numbers:

```
# iptables -n -L -v --line-numbers
```

Finally, same data output but related to INPUT/OUTPUT chains:

```
# iptables -L INPUT -n -viptables -L OUTPUT -n -v --line-numbers
```

List Rules as for a specific chain

```
# iptables -L INPUT
```

same data with rules specifications:

```
# iptables -S INPUT
```

rules list with packet count

```
# iptables -L INPUT -v
```

Delete/Insert rules

Delete Rule by Chain and Number

```
# iptables -D INPUT 10
```

Delete Rule by Specification

```
# iptables -D INPUT -m conntrack --ctstate INVALID -j DROP
```

Flush All Rules, Delete All Chains, and Accept All

```
# iptables -P INPUT ACCEPT
# iptables -P FORWARD ACCEPT
# iptables -P OUTPUT ACCEPT
# iptables -t nat -F
# iptables -t mangle -F
# iptables -F
# iptables -X
```

Flush All Chains

```
# iptables -F
```

Flush a Single Chain

```
# iptables -F INPUT
```

Insert Rule

```
# iptables -I INPUT 2 -s 202.54.1.2 -j DROP
```

Rules examples

Allow Loopback Connections

```
# iptables -A INPUT -i lo -j ACCEPTiptables -A OUTPUT -o lo -j ACCEPT
```

Allow Established and Related Incoming Connections

```
# iptables -A INPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
```

Allow Established Outgoing Connections

```
# iptables -A OUTPUT -m conntrack --ctstate ESTABLISHED -j ACCEPT
```

Internal to External

```
# iptables -A FORWARD -i eth1 -o eth0 -j ACCEPT
```

Drop Invalid Packets

```
# iptables -A INPUT -m conntrack --ctstate INVALID -j DROP
```

Block an IP Address

```
# iptables -A INPUT -s 192.168.1.10 -j DROP
```

Block and IP Address and Reject

```
# iptables -A INPUT -s 192.168.1.10 -j REJECT
```

Block Connections to a Network Interface

```
# iptables -A INPUT -i eth0 -s 192.168.1.10 -j DROP
```

Allow All Incoming SSH

```
# iptables -A INPUT -p tcp --dport 22 -m conntrack --ctstate NEW,ESTABLISHED -
j ACCEPT
# iptables -A OUTPUT -p tcp --sport 22 -m conntrack --ctstate ESTABLISHED -j
ACCEPT
```

Allow Incoming SSH from Specific IP address or subnet

```
# iptables -A INPUT -p tcp -s 192.168.1.0/24 --dport 22 -m conntrack --ctstate
NEW,ESTABLISHED -j ACCEPT
# iptables -A OUTPUT -p tcp --sport 22 -m conntrack --ctstate ESTABLISHED -j
ACCEPT
```

Allow Outgoing SSH

```
# iptables -A OUTPUT -p tcp --dport 22 -m conntrack --ctstate NEW,ESTABLISHED
-j ACCEPT
# iptables -A INPUT -p tcp --sport 22 -m conntrack --ctstate ESTABLISHED -j
ACCEPT
```

Allow Incoming Rsync from Specific IP Address or Subnet

```
# iptables -A INPUT -p tcp -s 192.168.1.0/24 --dport 873 -m conntrack --
ctstate NEW,ESTABLISHED -j ACCEPT
# iptables -A OUTPUT -p tcp --sport 873 -m conntrack --ctstate ESTABLISHED -j
ACCEPT
```

Allow Incoming HTTP

```
# iptables -A INPUT -p tcp --dport 80 -m conntrack --ctstate NEW,ESTABLISHED -
j ACCEPT
# iptables -A OUTPUT -p tcp --sport 80 -m conntrack --ctstate ESTABLISHED -j
ACCEPT
```

Allow Incoming HTTPS

```
# iptables -A INPUT -p tcp --dport 443 -m conntrack --ctstate NEW,ESTABLISHED
-j ACCEPT
# iptables -A OUTPUT -p tcp --sport 443 -m conntrack --ctstate ESTABLISHED -j
ACCEPT
```

Allow Incoming HTTP and HTTPS

```
# iptables -A INPUT -p tcp -m multiport --dports 80,443 -m conntrack --ctstate
NEW,ESTABLISHED -j ACCEPT
# iptables -A OUTPUT -p tcp -m multiport --dports 80,443 -m conntrack --
ctstate ESTABLISHED -j ACCEPT
```

Allow MySQL from Specific IP Address or Subnet

```
# iptables -A INPUT -p tcp -s 192.168.1.0/24 --dport 3306 -m conntrack --
ctstate NEW,ESTABLISHED -j ACCEPT
# iptables -A OUTPUT -p tcp --sport 3306 -m conntrack --ctstate ESTABLISHED -j
ACCEPT
```

Allow MySQL to Specific Network Interface

```
# iptables -A INPUT -i eth1 -p tcp --dport 3306 -m conntrack --ctstate
NEW,ESTABLISHED -j ACCEPT
# iptables -A OUTPUT -o eth1 -p tcp --sport 3306 -m conntrack --ctstate
ESTABLISHED -j ACCEPT
```

Allow PostgreSQL from Specific IP Address or Subnet

```
# iptables -A INPUT -p tcp -s 192.168.1.0/24 --dport 5432 -m conntrack --
ctstate NEW,ESTABLISHED -j ACCEPT
# iptables -A OUTPUT -p tcp --sport 5432 -m conntrack --ctstate ESTABLISHED -j
ACCEPT
```

Allow PostgreSQL to Specific Network Interface

```
# iptables -A INPUT -i eth1 -p tcp --dport 5432 -m conntrack --ctstate
NEW,ESTABLISHED -j ACCEPT
# iptables -A OUTPUT -o eth1 -p tcp --sport 5432 -m conntrack --ctstate
ESTABLISHED -j ACCEPT
```

Block Outgoing SMTP Mail

```
# iptables -A OUTPUT -p tcp --dport 25 -j REJECT
```

Allow All Incoming SMTP

```
# iptables -A INPUT -p tcp --dport 25 -m conntrack --ctstate NEW,ESTABLISHED -
j ACCEPT
```

```
# iptables -A OUTPUT -p tcp --sport 25 -m conntrack --ctstate ESTABLISHED -j
ACCEPT
```

Allow All Incoming IMAP

```
# iptables -A INPUT -p tcp --dport 143 -m conntrack --ctstate NEW,ESTABLISHED
-j ACCEPT
# iptables -A OUTPUT -p tcp --sport 143 -m conntrack --ctstate ESTABLISHED -j
ACCEPT
```

Allow All Incoming IMAPS

```
# iptables -A INPUT -p tcp --dport 993 -m conntrack --ctstate NEW,ESTABLISHED
-j ACCEPT
# iptables -A OUTPUT -p tcp --sport 993 -m conntrack --ctstate ESTABLISHED -j
ACCEPT
```

Allow All Incoming POP3

```
# iptables -A INPUT -p tcp --dport 110 -m conntrack --ctstate NEW,ESTABLISHED
-j ACCEPT
# iptables -A OUTPUT -p tcp --sport 110 -m conntrack --ctstate ESTABLISHED -j
ACCEPT
```

Allow All Incoming POP3S

```
# iptables -A INPUT -p tcp --dport 995 -m conntrack --ctstate NEW,ESTABLISHED
-j ACCEPT
# iptables -A OUTPUT -p tcp --sport 995 -m conntrack --ctstate ESTABLISHED -j
ACCEPT
```

Drop Private Network Address On Public Interface

```
# iptables -A INPUT -i eth1 -s 192.168.1.0/24 -j DROP
# iptables -A INPUT -i eth1 -s 10.0.0.0/8 -j DROP
```

Drop All Outgoing to Facebook Networks

Get Facebook AS:

```
# whois -h v4.whois.cymru.com " -v $(host facebook.com | grep "has address" |
cut -d " " -f4)" | tail -n1 | awk '{print $1}'
```

Drop:

```
# for i in $(whois -h whois.radb.net -- '-i origin AS1273' | grep "^route:" |
cut -d ":" -f2 | sed -e 's/^[ \t]*//' | sort -n -t . -k 1,1 -k 2,2 -k 3,3 -k
4,4 | cut -d ":" -f2 | sed 's/$/;/') ; do  iptables -A OUTPUT -s "$i" -j
REJECTdone
```

Log and Drop Packets

```
# iptables -A INPUT -i eth1 -s 10.0.0.0/8 -j LOG --log-prefix "IP_SPOOF A: "
# iptables -A INPUT -i eth1 -s 10.0.0.0/8 -j DROP
```

By default everything is logged to `/var/log/messages` file:

```
# tail -f /var/log/messagesgrep --color 'IP SPOOF' /var/log/messages
```

Log and Drop Packets with Limited Number of Log Entries

```
# iptables -A INPUT -i eth1 -s 10.0.0.0/8 -m limit --limit 5/m --limit-burst 7
-j LOG --log-prefix "IP_SPOOF A: "
# iptables -A INPUT -i eth1 -s 10.0.0.0/8 -j DROP
```

Drop or Accept Traffic From Mac Address

```
# iptables -A INPUT -m mac --mac-source 00:0F:EA:91:04:08 -j DROP
# iptables -A INPUT -p tcp --destination-port 22 -m mac --mac-source
00:0F:EA:91:04:07 -j ACCEPT
```

Block or Allow ICMP Ping Request

```
# iptables -A INPUT -p icmp --icmp-type echo-request -j DROP
# iptables -A INPUT -i eth1 -p icmp --icmp-type echo-request -j DROP
```

Specifying Multiple Ports with `multiport`

```
# iptables -A INPUT -i eth0 -p tcp -m state --state NEW -m multiport --dports
ssh,smtp,http,https -j ACCEPT
```

Load Balancing with `random*` or `nth*`

```
_ips=("172.31.250.10" "172.31.250.11" "172.31.250.12" "172.31.250.13")for ip
in "${_ips[@]}" ; do  iptables -A PREROUTING -i eth0 -p tcp --dport 80 -m
state --state NEW -m nth --counter 0 --every 4 --packet 0 \    -j DNAT --to-
destination ${ip}:80done
```

or

```
_ips=("172.31.250.10" "172.31.250.11" "172.31.250.12" "172.31.250.13")for ip
in "${_ips[@]}" ; do  iptables -A PREROUTING -i eth0 -p tcp --dport 80 -m
state --state NEW -m random --average 25 \    -j DNAT --to-destination
${ip}:80done
```

Restricting the Number of Connections with `limit` and `iplimit*`

```
# iptables -A FORWARD -m state --state NEW -p tcp -m multiport --dport
http,https -o eth0 -i eth1 -m limit --limit 20/hour --limit-burst 5 -j ACCEPT
```

or

```
# iptables -A INPUT -p tcp -m state --state NEW --dport http -m iplimit --
iplimit-above 5 -j DROP
```

Maintaining a List of recent Connections to Match Against

```
# iptables -A FORWARD -m recent --name portscan --rcheck --seconds 100 -j
DROPiptables -A FORWARD -p tcp -i eth0 --dport 443 -m recent --name portscan -
-set -j DROP
```

Matching Against a `string*` in a Packet's Data Payload

```
# iptables -A FORWARD -m string --string '.com' -j DROP
# iptables -A FORWARD -m string --string '.exe' -j DROP
```

Time-based Rules with `time*`

```
# iptables -A FORWARD -p tcp -m multiport --dport http,https -o eth0 -i eth1 -
m time --timestart 21:30 --timestop 22:30 --days Mon,Tue,Wed,Thu,Fri -j ACCEPT
```

Packet Matching Based on TTL Values

```
# iptables -A INPUT -s 1.2.3.4 -m ttl --ttl-lt 40 -j REJECT
```

Protection against port scanning

```
# iptables -N port-scanningiptables -A port-scanning -p tcp --tcp-flags
SYN,ACK,FIN,RST RST -m limit --limit 1/s --limit-burst 2 -j RETURNiptables -A
port-scanning -j DROP
```

SSH brute-force protection

```
# iptables -A INPUT -p tcp --dport ssh -m conntrack --ctstate NEW -m recent --
setiptables -A INPUT -p tcp --dport ssh -m conntrack --ctstate NEW -m recent -
```

```
-update --seconds 60 --hitcount 10 -j DROP
```

Syn-flood protection

```
# iptables -N syn_floodiptables -A INPUT -p tcp --syn -j syn_floodiptables -A
syn_flood -m limit --limit 1/s --limit-burst 3 -j RETURN
# iptables -A syn_flood -j DROPiptables -A INPUT -p icmp -m limit --limit  1/s
--limit-burst 1 -j ACCEPT
# iptables -A INPUT -p icmp -m limit --limit 1/s --limit-burst 1 -j LOG --log-
prefix PING-DROP:
# iptables -A INPUT -p icmp -j DROPiptables -A OUTPUT -p icmp -j ACCEPT
```

Mitigating SYN Floods With SYNPROXY

```
# iptables -t raw -A PREROUTING -p tcp -m tcp --syn -j CT --notrack
# iptables -A INPUT -p tcp -m tcp -m conntrack --ctstate INVALID,UNTRACKED -j
SYNPROXY --sack-perm --timestamp --wscale 7 --mss 1460
# iptables -A INPUT -m conntrack --ctstate INVALID -j DROP
```

Block New Packets That Are Not SYN

```
# iptables -A INPUT -p tcp ! --syn -m state --state NEW -j DROP
```

or

```
# iptables -t mangle -A PREROUTING -p tcp ! --syn -m conntrack --ctstate NEW -
j DROP
```

Force Fragments packets check

```
# iptables -A INPUT -f -j DROP
```

XMAS packets

```
# iptables -A INPUT -p tcp --tcp-flags ALL ALL -j DROP
```

Drop all NULL packets

```
# iptables -A INPUT -p tcp --tcp-flags ALL NONE -j DROP
```

Block Uncommon MSS Values

```
# iptables -t mangle -A PREROUTING -p tcp -m conntrack --ctstate NEW -m tcpmss
! --mss 536:65535 -j DROP
```

Block Packets With Bogus TCP Flags

```
# iptables -t mangle -A PREROUTING -p tcp --tcp-flags FIN,SYN,RST,PSH,ACK,URG
NONE -j DROP
# iptables -t mangle -A PREROUTING -p tcp --tcp-flags FIN,SYN FIN,SYN -j DROP
# iptables -t mangle -A PREROUTING -p tcp --tcp-flags SYN,RST SYN,RST -j DROP
# iptables -t mangle -A PREROUTING -p tcp --tcp-flags FIN,RST FIN,RST -j DROP
# iptables -t mangle -A PREROUTING -p tcp --tcp-flags FIN,ACK FIN -j DROP
# iptables -t mangle -A PREROUTING -p tcp --tcp-flags ACK,URG URG -j DROP
# iptables -t mangle -A PREROUTING -p tcp --tcp-flags ACK,FIN FIN -j DROP
```

```
# iptables -t mangle -A PREROUTING -p tcp --tcp-flags ACK,PSH PSH -j DROP
# iptables -t mangle -A PREROUTING -p tcp --tcp-flags ALL ALL -j DROP
# iptables -t mangle -A PREROUTING -p tcp --tcp-flags ALL NONE -j DROP
# iptables -t mangle -A PREROUTING -p tcp --tcp-flags ALL FIN,PSH,URG -j DROP
# iptables -t mangle -A PREROUTING -p tcp --tcp-flags ALL SYN,FIN,PSH,URG -j
DROP
# iptables -t mangle -A PREROUTING -p tcp --tcp-flags ALL SYN,RST,ACK,FIN,URG
-j DROP
```

Block Packets From Private Subnets (Spoofing)

```
_subnets=("224.0.0.0/3" "169.254.0.0/16" "172.16.0.0/12" "192.0.2.0/24"
"192.168.0.0/16" "10.0.0.0/8" "0.0.0.0/8" "240.0.0.0/5")for _sub in
"${_subnets[@]}" ; do  iptables -t mangle -A PREROUTING -s "$_sub" -j
DROPdoneiptables -t mangle -A PREROUTING -s 127.0.0.0/8 ! -i lo -j DROP
```

Saving Rules

On Debian Based systems:

```
# netfilter-persistent save
```

On RedHat Based systems

```
# service iptables save
```