Linux Firewalls

IpTables (all distributions):

Iptables consists of rules in tables in chains. The chains are INPUT OUTPUT and FORWARD.

`iptables -S` Print the current rules

`iptables -F` Flush the tables, deleting all rules

The following is a iptables rule file

`Vim /etc/iptables.rules`

```
*filter

# Allows all loopback traffic and drops traffic to 127/8 that
doesn't use the loopback interface
-A INPUT -i lo -j ACCEPT
-A OUTPUT -o lo -j ACCEPT
-A INPUT -d 127.0.0.0/8 -j DROP
-A OUTPUT -d 127.0.0.0/8 -j DROP

# Accepts established inbound connections
-A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
-A OUTPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

# Protect from malformed pings
-N BAD_PING
-A BAD_PING -p icmp --icmp-type echo-request -m hashlimit
--hashlimit 1/s --hashlimit-burst 10 --hashlimit-htable-expire
300000 --hashlimit-mode srcip --hashlimit-name t_BAD_PING -j RETURN
-A BAD_PING -j DROP
-A INPUT -p icmp --icmp-type echo-request -j BAD_PING

# Prevent port scanning
-N PORTSCAN
-A PORTSCAN -p tcp --tcp-flags ACK,FIN FIN -j DROP
-A PORTSCAN -p tcp --tcp-flags ACK,PSH PSH -j DROP
-A PORTSCAN -p tcp --tcp-flags ACK,URG URG -j DROP
-A PORTSCAN -p tcp --tcp-flags FIN,RST FIN,RST -j DROP
-A PORTSCAN -p tcp --tcp-flags SYN,FIN SYN,FIN -j DROP
```

```
-A PORTSCAN -p tcp --tcp-flags SYN,RST SYN,RST -j DROP
-A PORTSCAN -p tcp --tcp-flags ALL ALL -j DROP
-A PORTSCAN -p tcp --tcp-flags ALL NONE -j DROP
-A PORTSCAN -p tcp --tcp-flags ALL FIN,PSH,URG -j DROP
-A PORTSCAN -p tcp --tcp-flags ALL SYN,FIN,PSH,URG -j DROP
-A PORTSCAN -p tcp --tcp-flags ALL SYN,FIN,PSH,URG -j DROP
-A PORTSCAN -p tcp --tcp-flags ALL SYN,RST,ACK,FIN,URG -j DROP

# Drop fragmented packets
-A INPUT -f -j DROP
-A INPUT -p tcp ! --syn -m state --state NEW -j DROP

# Allow dns queries
-A INPUT -p udp --sport 53 -j ACCEPT
-A OUTPUT -p udp --dport 53 -j ACCEPT

# Open tcp ports for incoming traffic
# ssh
-A INPUT -p tcp -m state --state NEW,ESTABLISHED --dport 2222 -j
ACCEPT
-A OUTPUT -p tcp -m state --state ESTABLISHED --sport 2222 -j ACCEPT
# web traffic
-A INPUT -p tcp --dport 80 -m state --state NEW,ESTABLISHED -j
ACCEPT
-A OUTPUT -p tcp --sport 80 -m state --state ESTABLISHED -j ACCEPT
-A INPUT -p tcp --dport 443 -m state --state NEW,ESTABLISHED -j
ACCEPT
-A OUTPUT -p tcp --sport 443 -m state --state ESTABLISHED -j ACCEPT

# Open tcp ports for outgoing traffic
-A INPUT -p tcp --sport 80 -m state --state ESTABLISHED -j ACCEPT
-A OUTPUT -p tcp --dport 80 -m state --state NEW,ESTABLISHED -j
ACCEPT
-A INPUT -p tcp --sport 443 -m state --state ESTABLISHED -j ACCEPT
-A OUTPUT -p tcp --dport 443 -m state --state NEW,ESTABLISHED -j
ACCEPT

# Logging
```

```
-A INPUT -m limit --limit 5/min -j LOG --log-prefix "iptables
denied: " --log-level 7

# Default block everything else
-A INPUT -j DROP
-A FORWARD -j DROP
-A OUTPUT -j ACCEPT

COMMIT
```

`Iptables-restore` *`iptables_filename`* load the iptables ruleset
The configuration above will log rejected packets, to put all logged packets into one file edit /etc/rsyslog.d/*filename*

```
:msg, contains, "iptables denied: " /var/log/iptables.log
& ~
```

IMPORTANT
Iptables-restore is not permanent, each reboot the command must be rerun unless a startup script is created or the iptables-persistent package is installed. Making a script to have it automatically apply the iptables rules via script simply add a script to

```
Vim /etc/network/if-pre-up.d/iptables
      #!/bin/bash
      /sbin/iptables-restore < /etc/iptables.rules
```

Centos/RHEL:

Firewalld-

`yum install firewalld` Install firewalld
`firewall-cmd --state` Check the status of the firewall
`firewall-cmd --list-all` Check what ports are open
`firewall-cmd --list-services` lists services currently allowed through firewall
`firewall-cmd --add-port=# or service/tcp or udp --permanent` Open a port or service permanently
`Firewall-cmd --remove-port=# or service/tcp or udp --permanent` Closes an open port or service permanently
`firewall-cmd --reload` reloads firewall rules after adding or removing ports
`systemctl start firewall-cmd` Starts the firewall-cmd service
`systemctl stop firewall-cmd` Stops the firewall-cmd service
`systemctl restart firewall-cmd` Restarts the firewall-cmd service
`systemctl enable firewall-cmd` Enables the firewall-cmd service to start automatically after reboot

`Firewall-cmd --panic-on` All packets in both directions are dropped and all connections are severed

`Firewall-cmd --panic-off` Disables panic mode. If panic was on for a short period, active connections may revive, though not guaranteed

Ubuntu/Debian:

UFW-

`Ufw enable` enables the firewall on boot and reloads it

`Ufw disable` unloads the firewall and disables it on boot

`Ufw default allow {direction}` default policy for direction **Incoming Outgoing** or **Routed**

`Ufw allow {port number}/{protocol}` opens the port to protocol (tcp or udp)

`Ufw deny {port number}/{protocol}` closes the port to protocol (tcp or udp)

`Ufw status` shows the status of the firewall

`Ufw reload` refreshes the firewall with updated rules

`Ufw reset` disables firewall and resets to the default firewall configuration

`Ufw logging on {level}` turns on logging and uses a syslog facility level. Logs are saved to `/var/log/ufw.log` default level is low

`Ufw allow in from {ip}`

`Ufw limit {service}/{protocol}`