Replace SuSE Firewall with /etc/sysconfig/iptables

23.02.25

Pegasi Knowledge
https://ghost.pegasi.fi/wiki/

Table of Contents

Replace SuSE Firewall with /etc/sysconfig/iptables

Updated 09.03.2020 for OpenSuse Leap 15.1. I needed to allow all access from my local virtual guest to my Linux box. Why an earth would I go through multiple steps and custom scripts when I can accomplish my access with a one simple line in /etc/sysconfig/iptables? All in all my iptables config is just a few lines, compared to multiple files and custom scripts of SuSEFirewall.

I tried to say it in a decent manner but I just cannot keep this inside me. SuSEfirewall is just terrible. But that is just me who has used /etc/sysconfig/iptables since its arrival to CentOS / RHEL. I am sure it manages to do well when in client use but still I think it is important to learn to use iptables so you know what you're really doing. And if you're ever into servers then you must do it like this anyway.

Disabled SuSEfirewall and install iptables

```
systemctl stop firewalld
systemctl disable firewalld
zypper in iptables
```

Set up basic firewall rules

Open iptables configuration file

```
vim /etc/sysconfig/iptables
```

Put basic stuff inside to make your Linux safe. Example content would be:

```
*filter
:INPUT DROP [0:0]
:FORWARD DROP [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -s 127.0.0.1/32 -j ACCEPT
-A INPUT -j LOG
COMMIT
```

Put the rules in effect:

```
iptables-restore /etc/sysconfig/iptables
```

Add or remove rules afterwards

Add and save new rule from command line for example add access with ssh from 1.2.3.4:

```
iptables -I INPUT -p tcp -m tcp -s 1.2.3.4 --dport 22 -j ACCEPT
iptables-save > /etc/sysconfig/iptables
```

Remove is easiest to do by removing the corresponding line from config file and running:

```
iptables-restore /etc/sysconfig/iptables
```

Make it persistant

Is it persistant or persistent? :)

First lets make a systemd configuration file /usr/lib/systemd/system/iptables.service with contents:

```
[Unit]
Description=IPv4 firewall with iptables
After=syslog.target
Before=ip6tables.service
AssertPathExists=/etc/sysconfig/iptables

[Service]
Type=oneshot
RemainAfterExit=yes
ExecStart=/usr/sbin/iptables-restore /etc/sysconfig/iptables
ExecReload=/usr/sbin/iptables-restore /etc/sysconfig/iptables
ExecStop=/usr/sbin/iptables -F
Environment=BOOTUP=serial
Environment=CONSOLETYPE=serial
StandardOutput=syslog
StandardError=syslog

[Install]
WantedBy=basic.target
```

The systemd script above does not properly flush all of the iptables rules when using systemctl stop iptables but I do not want it to. If you need to do it make a script which issues the following commands:

```
iptables -P INPUT ACCEPT
iptables -P FORWARD ACCEPT
iptables -P OUTPUT ACCEPT
```

```
iptables -t nat -F
iptables -t mangle -F
iptables -F
iptables -X
```

And put it to the script above as stop command.

Finally we enable the service:

```
systemctl enable iptables
```

Now with every reboot we have iptable rules loaded.