FirewallD

Introduction

FirewallD is frontend controller for iptables used to implement persistent network traffic rules. It provides command line and graphical interfaces and is available in the repositories of most Linux distributions. Working with FirewallD has two main differences compared to directly controlling iptables:

1. FirewallD uses zones and services instead of chain and rules.
2. It manages rulesets dynamically, allowing updates without breaking existing sessions and connections.

> FirewallD is a wrapper for iptables to allow easier management of iptables rules–it is not an iptables replacement. While iptables commands are still available to FirewallD, it's recommended to use only FirewallD commands with FirewallD.

Install

```
#
sudo yum install firewalld     # [CentOS 7/RHEL 7]
sudo dnf install firewalld     # [CentOS 8/RHEL 8/Fedora]
sudo zypper install firewalld  # [openSUSE Leap]

# Autostart the service
systemctl enable firewalld
systemctl restart firewalld
```

Firewalld Zones

```
$ firewall-cmd --get-zones

block dmz drop external home internal public trusted work
```

- **block**: Any incoming connections are rejected with an icmp-host-prohibited message for IPv4 and icmp6-adm-prohibited for IPv6. Only network connections initiated within this system are allowed.
- **dmz**: Used for computers located in your demilitarized zone that are publicly-accessible with limited access to your internal network. Only selected incoming connections are accepted.
- **drop**: Any incoming connections are dropped without any notification. Only outgoing connections are allowed.
- **external**: For use on external networks with NAT masquerading enabled when your system acts as a router. Only selected incoming connections are allowed.
- **home**: Used for home network and other computers on the same networks are mostly trusted. Only selected incoming connections are accepted.
- **internal**: For use on internal networks, and other systems on the network are generally trusted. Only selected incoming connections are accepted.
- **public**: For use in public areas, but you should not trust the other computers on networks. Only selected incoming connections are accepted.
- **trusted**: All network connections are accepted.
- **work**: For use in work areas, and other computers on the same networks are mostly trusted. Only selected incoming connections are accepted.

Firewalld Services

```
$ firewall-cmd --get-services


RH-Satellite-6 amanda-client amanda-k5-client amqp amqps apcupsd bacula bacula-client bgp..
```

Firewalld Runtime and Permanent Settings

Firewalld uses two separate configurations namely runtime, and permanent:

- **Runtime** Configuration: The runtime configuration will not be persistent on system reboots, and the firewalld service stop. It means the runtime configuration are not automatically saved to the permanent configuration.
- **Permanent** Configuration: The permanent configuration is stored in configuration files and will be loaded and becomes a new runtime configuration across every reboot or service reload/restart. Note that, to make the changes permanent you need to use the –permanent option with firewall-cmd.

Enabling Firewalld

```
$ sudo systemctl start firewalld

$ sudo systemctl enable firewalld


# To check the status of firewalld
```

```
$ sudo firewall-cmd --state
```

Zone Management

```
# To view the default zone
firewall-cmd --get-default-zone

# To view the zone configuration of the default zone
firewall-cmd --list-all

# to assign the ‘eth0’ interface to ‘home’ zone
firewall-cmd --zone=home --change-interface=eth0

# To view all active zones
firewall-cmd --get-active-zones

# To change the default zone
firewall-cmd --set-default-zone=home

# To print the specific zone configuration
firewall-cmd --zone=home --list-all

# To get a list of all the available zones
firewall-cmd --get-zones

# To find out which zone is associated with the eth0 interface
firewall-cmd --get-zone-of-interface=eth0

# To create a new zone
firewall-cmd --permanent --new-zone=daygeek

# To migrate runtime settings to permanent
firewall-cmd --runtime-to-permanent
```

How to use

???????? zone??? zone ?? public?

?? public zone ????

```
firewall-cmd --list-all-zones
```

```
...
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: ens160
  sources:
  services: ssh dhcpv6-client
  ports:
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
...
```

??????

```
# verify the default config and zones
firewall-cmd --get-default-zone

# List information for all zones
firewall-cmd --list-all-zones

# List allowed  services
firewall-cmd --zone=work --list-services

# Remove the SSH service from the default zone ( public)
firewall-cmd --permanent --remove-service=ssh

# Create the zone, allow the SSH service and the source IPs
firewall-cmd --permanent --new-zone=SSHZONE
firewall-cmd --permanent --zone=SSHZONE --add-source=[I.P.]
firewall-cmd --permanent --zone=SSHZONE --add-service=ssh
firewall-cmd --permanent --zone=grafana --add-port=3000/tcp
# remove the port
firewall-cmd --permanent --zone=grafana --remove-port=80/tcp
```

```

# Reload the firewall to take effect and make the zone active
firewall-cmd --reload
```

Example#1: Public zone by default

?? public ??? ssh ? dhcp-client

```
# □□□□ IP □□□□□□□
firewall-cmd --permanent --zone=trusted --add-source=10.18.109.20
firewall-cmd --reload
firewall-cmd --zone=trusted --list-all
```

Example#2: Add an IP to allow the access to the port 3000

```
firewall-cmd --permanent --new-zone=grafana
firewall-cmd --permanent --zone=grafana --add-port=3000/tcp
firewall-cmd --permanent --zone=grafana --add-source=10.18.109.20
firewall-cmd --permanent --list-all --zone=grafana
firewall-cmd --reload
```

Example#3: Remove an IP from specified zone.

```
firewall-cmd --zone=grafana --remove-source=10.18.109.20
firewall-cmd --runtime-to-permanent
```

Example#4: Allow all IPs to access to the port 3000

```
firewall-cmd --permanent --zone=public --add-port=3000/tcp
firewall-cmd --reload

# □□□□□□□ port 6443,2379,2380,10250
firewall-cmd --zone=public --permanent --add-port={6443,2379,2380,10250}/tcp
firewall-cmd --reload
firewall-cmd --list-ports
```

Firewalld logging

```
sudo firewall-cmd --get-log-denied
sudo firewall-cmd --set-log-denied=all
sudo firewall-cmd --get-log-denied
```

View denied packets

```
journalctl -x -e
```

```
sudo systemctl restart rsyslog.service
sudo tail -f /var/log/firewalld-droppd.log
```

log all dropped packets to /var/log/firewalld-droppd.log file

```
sudo vim /etc/rsyslog.d/firewalld-droppd.conf
```

```
:msg,contains,"_DROP" /var/log/firewalld-droppd.log
:msg,contains,"_REJECT" /var/log/firewalld-droppd.log
& stop
```

```
sudo systemctl restart rsyslog.service
sudo tail -f /var/log/firewalld-droppd.log
```

Tutorials

- A beginner's guide to firewalld in Linux
- How to set up a firewall using FirewallD on CentOS 8
- Introduction to FirewallD on CentOS
- How to Configure ‘FirewallD’ in RHEL/CentOS 7 and Fedora 21
- How to Open Port for a Specific IP Address in Firewalld
- How to configure firewalld rules in Linux

Revision #23
Created 4 September 2020 19:03:59 by Admin
Updated 20 May 2023 09:43:45 by Admin