Introduction to firewalld on CentOS

Updated Wednesday, October 7, 2020 , by Linode Community

Bottom of Form

[firewalld](http://www.firewalld.org/) is frontend controller for iptables. 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.

**Note**

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.

**Installing and Managing firewalld**

1. Check the firewall status. The output should say either running or not running.

sudo firewall-cmd --state

1. To view the status of the firewalld daemon:

sudo systemctl status firewalld

1. To reload a firewalld configuration:

sudo firewall-cmd --reload

**Configuring firewalld**

firewalld is configured with XML files. Except for very specific configurations, you won’t have to deal with them and **firewall-cmd** should be used instead.

Configuration files are located in two directories:

* /usr/lib/firewalld holds default configurations like default zones and common services. Avoid updating them because those files will be overwritten by each firewalld package update.
* /etc/firewalld holds system configuration files. These files will overwrite a default configuration.

**Configuration Sets**

firewalld uses two *configuration sets*: Runtime and Permanent. Runtime configuration changes are not retained on reboot or upon restarting firewalld whereas permanent changes are not applied to a running system.

By default, firewall-cmd commands apply to runtime configuration but using the --permanent flag will establish a persistent configuration. To add and activate a permanent rule, you can use one of two methods.

1. Add the rule to both the permanent and runtime sets.

sudo firewall-cmd --zone=public --add-service=http --permanent

sudo firewall-cmd --zone=public --add-service=http

1. Add the rule to the permanent set and reload firewalld.

sudo firewall-cmd --zone=public --add-service=http --permanent

sudo firewall-cmd --reload

**Note**

The reload command drops all runtime configurations and applies a permanent configuration. Because firewalld manages the ruleset dynamically, it won’t break an existing connection and session.

**Firewall Zones**

Zones are just pre-configured rulesets for various trust levels you would likely have for a given location or scenario (e.g. home, public, trusted, etc.). Different zones allow different network services and incoming traffic types while denying everything else. After enabling firewalld for the first time, *Public* will be the default zone.

Zones can also be applied to different network interfaces. For example, with separate interfaces for both an internal network and the Internet, you can allow DHCP on an internal zone but only HTTP and SSH on external zone. Any interface not explicitly set to a specific zone will be attached to the default zone.

To view the default zone:

sudo firewall-cmd --get-default-zone

To change the default zone:

sudo firewall-cmd --set-default-zone=internal

To see the zones used by your network interface(s):

sudo firewall-cmd --get-active-zones

Example output:

public

interfaces: eth0

To get all configurations for a specific zone:

sudo firewall-cmd --zone=public --list-all

Example output:

public (active)

target: default

icmp-block-inversion: no

interfaces: eth0

sources:

services: ssh dhcpv6-client http

ports: 12345/tcp

protocols:

masquerade: no

forward-ports:

source-ports:

icmp-blocks:

rich rules:

To get all configurations for all zones:

sudo firewall-cmd --list-all-zones

Example output:

trusted

target: ACCEPT

icmp-block-inversion: no

interfaces:

sources:

services:

ports:

protocols:

masquerade: no

forward-ports:

source-ports:

icmp-blocks:

rich rules:

...

work

target: default

icmp-block-inversion: no

interfaces:

sources:

services: ssh dhcpv6-client

ports:

protocols:

masquerade: no

forward-ports:

source-ports:

icmp-blocks:

rich rules:

**Working with Services**

firewalld can allow traffic based on predefined rules for specific network services. You can create your own custom service rules and add them to any zone. The configuration files for the default supported services are located at /usr/lib/firewalld/services and user-created service files would be in /etc/firewalld/services.

To view the default available services:

sudo firewall-cmd --get-services

As an example, to enable or disable the HTTP service:

sudo firewall-cmd --zone=public --add-service=http --permanent

sudo firewall-cmd --zone=public --remove-service=http --permanent

**Allowing or Denying an Arbitrary Port/Protocol**

As an example: Allow or disable TCP traffic on port 12345.

sudo firewall-cmd --zone=public --add-port=12345/tcp --permanent

sudo firewall-cmd --zone=public --remove-port=12345/tcp --permanent

**Port Forwarding**

The example rule below forwards traffic from port 80 to port 12345 on **the same server**.

sudo firewall-cmd --zone="public" --add-forward-port=port=80:proto=tcp:toport=12345

To forward a port to **a different server**:

1. Activate masquerade in the desired zone.

sudo firewall-cmd --zone=public --add-masquerade

1. Add the forward rule. This example forwards traffic from local port 80 to port 8080 on *a remote server* located at the IP address: 198.51.100.0.

sudo firewall-cmd --zone="public" --add-forward-port=port=80:proto=tcp:toport=8080:toaddr=198.51.100.0

To remove the rules, substitute --add with --remove. For example:

sudo firewall-cmd --zone=public --remove-masquerade

**Constructing a Ruleset with firewalld**

As an example, here is how you would use firewalld to assign basic rules to your Linode if you were running a web server.

1. Assign the *dmz* zone as the default zone to eth0. Of the default zones offered, dmz (demilitarized zone) is the most desirable to start with for this application because it allows only SSH and ICMP.

sudo firewall-cmd --set-default-zone=dmz

sudo firewall-cmd --zone=dmz --add-interface=eth0

1. Add permanent service rules for HTTP and HTTPS to the dmz zone:

sudo firewall-cmd --zone=dmz --add-service=http --permanent

sudo firewall-cmd --zone=dmz --add-service=https --permanent

1. Reload firewalld so the rules take effect immediately:

sudo firewall-cmd --reload

If you now run firewall-cmd --zone=dmz --list-all, this should be the output:

dmz (default)

interfaces: eth0

sources:

services: http https ssh

ports:

masquerade: no

forward-ports:

icmp-blocks:

rich rules:

This tells us that the **dmz** zone is our **default** which applies to the **eth0 interface**, all network **sources** and **ports**. Incoming HTTP (port 80), HTTPS (port 443) and SSH (port 22) traffic is allowed and since there are no restrictions on IP versioning, this will apply to both IPv4 and IPv6. **Masquerading** and **port forwarding** are not allowed. We have no **ICMP blocks**, so ICMP traffic is fully allowed, and no **rich rules**. All outgoing traffic is allowed.

**Advanced Configuration**

Services and ports are fine for basic configuration but may be too limiting for advanced scenarios. Rich Rules and Direct Interface allow you to add fully custom firewall rules to any zone for any port, protocol, address and action.

**Rich Rules**

Rich rules syntax is extensive but fully documented in the [firewalld.richlanguage(5)](https://jpopelka.fedorapeople.org/firewalld/doc/firewalld.richlanguage.html" \t "_blank) man page (or see man firewalld.richlanguage in your terminal). Use --add-rich-rule, --list-rich-rules and --remove-rich-rule with firewall-cmd command to manage them.

Here are some common examples:

Allow all IPv4 traffic from host 192.0.2.0.

sudo firewall-cmd --zone=public --add-rich-rule 'rule family="ipv4" source address=192.0.2.0 accept'

Deny IPv4 traffic over TCP from host 192.0.2.0 to port 22.

sudo firewall-cmd --zone=public --add-rich-rule 'rule family="ipv4" source address="192.0.2.0" port port=22 protocol=tcp reject'

Allow IPv4 traffic over TCP from host 192.0.2.0 to port 80, and forward it locally to port 6532.

sudo firewall-cmd --zone=public --add-rich-rule 'rule family=ipv4 source address=192.0.2.0 forward-port port=80 protocol=tcp to-port=6532'

Forward all IPv4 traffic on port 80 to port 8080 on host 198.51.100.0 (masquerade should be active on the zone).

sudo firewall-cmd --zone=public --add-rich-rule 'rule family=ipv4 forward-port port=80 protocol=tcp to-port=8080 to-addr=198.51.100.0'

To list your current Rich Rules in the public zone:

sudo firewall-cmd --zone=public --list-rich-rules

**iptables Direct Interface**

For the most advanced usage, or for iptables experts, firewalld provides a direct interface that allows you to pass raw iptables commands to it. Direct Interface rules are not persistent unless the --permanent is used.

To see all custom chains or rules added to firewalld:

firewall-cmd --direct --get-all-chains

firewall-cmd --direct --get-all-rules

Discussing iptables syntax details goes beyond the scope of this guide. If you want to learn more, you can review our [iptables guide](https://www.linode.com/docs/security/firewalls/control-network-traffic-with-iptables/).