



Red Hat Enterprise Linux 7

Firewalld HowTo

Patrick Ladd
Technical Account Manager,
Red Hat
pladd@redhat.com

What Is firewalld?

- Dynamic, modern control of system firewall functions
- Still iptables underneath
- Major features;
 - Real time rule changes without interruption
 - Zones to simplify and segregate configuration
 - Separate network traffic & rules by interface and zone
 - GUI that works
 - System configs in `/usr/lib/firewalld/*`
 - Custom configs in `/etc/firewalld/*`
 - Daemon runs in user space
 - Protocol independent: IPv4 & IPv6

Zones

- Manages groups of rules
- Dictate what traffic should be allowed
 - Based on level of trust in connected network(s)
 - Based on origin of packet
- Network interfaces are assigned a zone

Default Pre-Defined Zones

- **drop**

Drop all incoming traffic unless related to outgoing traffic (do not even respond with ICMP errors).

- **block**

Reject all incoming traffic unless related to outgoing traffic.

- **dmz**

Reject incoming traffic unless related to outgoing traffic or matching the ssh pre-defined service.

- **external**

Reject incoming traffic unless related to outgoing traffic or matching the ssh pre-defined service. Outgoing IPv4 traffic forwarded through this zone is masqueraded to look like it originated from the IPv4 address of the outgoing network interface.

Default Pre-Defined Zones

- **public**

Reject incoming traffic unless related to outgoing traffic or matching the ssh, or dhcpv6-client pre-defined services. The default zone for newly-added network interfaces.

- **work**

Reject incoming traffic unless related to outgoing traffic or matching the ssh, ipp-client, or dhcpv6-client pre-defined services.

Default Pre-Defined Zones

- **internal**

Reject incoming traffic unless related to outgoing traffic or matching the ssh, mdns, ipp-client, samba-client, or dhcpv6-client pre-defined services.

- **home**

Same as internal

- **trusted**

Allow all incoming traffic.

Turning on firewalld

```
$ sudo systemctl start firewalld.service
```

We can verify that the service is running and reachable by typing:

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

Output:

```
running
```

This indicates that our firewall is up and running with the default configuration.

Exploring Zones

We can see which zone is currently selected as the default by typing:

```
$ firewall-cmd --get-default-zone
```

output

```
public
```

Since we haven't given firewalld any commands to deviate from the default zone, and none of our interfaces are configured to bind to another zone, that zone will also be the only "active" zone (the zone that is controlling the traffic for our interfaces). We can verify that by typing:

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

output

```
public
```

```
interfaces: eth0 eth1
```

Here, we can see that we have two network interfaces being controlled by the firewall (eth0 and eth1). They are both currently being managed according to the rules defined for the public zone.

Examining Alternate Zones

Now we have a good idea about the configuration for the default and active zone. We can find out information about other zones as well.

To get a list of the available zones, type:

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

output

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

Examining Alternate Zones

We can see the specific configuration associated with a zone by including the `--zone=` parameter in our `--list-all` command:

```
$ firewall-cmd --zone=home --list-all
```

output

home

interfaces:

sources:

services: dhcpv6-client ipp-client mdns samba-client ssh

ports:

masquerade: no

forward-ports:

icmp-blocks:

rich rules:

You can output all of the zone definitions by using the `--list-all-zones` option. You will probably want to pipe the output into a pager for easier viewing:

```
$ firewall-cmd --list-all-zones | less
```

Changing Zones

You can transition an interface between zones during a session by using the `--zone=` parameter in combination with the `--change-interface=` parameter. As with all commands that modify the firewall, you will need to use `sudo`.

For instance, we can transition our `eth0` interface to the "home" zone by typing this:

```
$ sudo firewall-cmd --zone=home --change-interface=eth0
```

output

success

Checking Zones

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

output

home

```
  interfaces: eth0
```

public

```
  interfaces: eth1
```

If the firewall is completely restarted, the interface will revert to the default zone:

```
$ sudo systemctl restart firewalld.service
```

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

output

public

```
  interfaces: eth0 eth1
```

Setting the Default Zone

If all of your interfaces can best be handled by a single zone, it's probably easier to just select the best default zone and then use that for your configuration.

You can change the default zone with the `--set-default-zone=` parameter. This will immediately change any interface that had fallen back on the default to the new zone:

```
$ sudo firewall-cmd --set-default-zone=home
```

output

home

```
interfaces: eth0 eth1
```

Making Permanent Zone Changes

- Modify the zone for the connection in network manager
 - `nmcli conn modify <iface> connection.zone <zone>`
- Modify zone for the connection in firewalld
 - `firewall-cmd --zone=home --change-interface=eth0 --permanent`
- Optional - restart the network and firewalld service
 - `systemctl restart network.service`

Making Changes Permanent

- Any zone or rule change can be designated as either **permanent** or **immediate**.
- If a parameter is added or modified, by default, the behavior of the **currently running firewall** is modified.
- At the next boot, the old rules will be reverted.
- Most firewall-cmd operations can take the `--permanent` flag to indicate that the non-ephemeral firewall should be targeted.
- This will affect the rule set that is reloaded upon boot. This separation means that you can test rules in your active firewall instance and then reload if there are problems.

Firewall Services

Services are sets of firewall rules to open ports associated with a particular application or system service.

The easiest method is to add the services or ports you need to the zones you are using. Again, you can get a list of the available services with the `--get-services` option:

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

output

```
RH-Satellite-6 amanda-client bacula bacula-client dhcp  
dhcpv6 dhcpv6-client dns ftp high-availability http https  
imaps ipp ipp-client ipsec kerberos kpasswd ldap ldaps  
libvirt libvirt-tls mdns mntd ms-wbt mysql nfs ntp openvpn  
pmcd pmproxy pmwebapi pmwebapis pop3s postgresql proxy-dhcp  
radius rpc-bind samba samba-client smtp ssh telnet tftp  
tftp-client transmission-client vnc-server wbem-https
```


Firewall Service Definitions

Each service is defined by its associated .xml file within the `/usr/lib/firewalld/services` directory. For instance, the SSH service is defined like this:

```
/usr/lib/firewalld/services/ssh.xml
```

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<service>
```

```
  <short>SSH</short>
```

```
  <description>Secure Shell (SSH) is a protocol for logging into and
executing commands on remote machines. It provides secure encrypted
communications. If you plan on accessing your machine remotely via
SSH over a firewalled interface, enable this option. You need the
openssh-server package installed for this option to be
useful.</description>
```

```
  <port protocol="tcp" port="22"/>
```

```
</service>
```

You can create your own service definitions. The easiest way is to copy & edit an existing service definition. The service you create will take the name of the .xml definition file.

Activating Services

You can enable a service for a zone using the `--add-service=` parameter. The operation will target the default zone or whatever zone is specified by the `--zone=` parameter.

For instance, if we are running a web server serving conventional HTTP traffic, we can allow this traffic for interfaces in our "public" zone for this session by typing:

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

You can leave out the `--zone=` if you wish to modify the default zone. We can verify the operation was successful by using the `--list-all` or `--list-services` operations:

```
$ firewall-cmd --zone=public --list-services
```

output

```
dhcpv6-client http ssh
```

Activating Services... Permanently

Once you have tested that everything is working as it should, you will probably want to modify the permanent firewall rules so that your service will still be available after a reboot. We can make our "public" zone change permanent by typing:

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

You can verify that this was successful by adding the `--permanent` flag to the `--list-services` operation. **You need to be root for any `--permanent` operations:**

```
$ sudo firewall-cmd --zone=public --permanent --list-services
```

output

```
dhcpv6-client http ssh
```

Individual Firewall Rules

If you do not wish to create a new service or modify an existing one, you can open up individual ports in your firewall on an individual basis.

For instance, if our application runs on port 5000 and uses TCP, we could add this to the "public" zone for this session using the `--add-port=` parameter. Protocols can be either `tcp` or `udp`:

```
$ sudo firewall-cmd --zone=public --add-port=5000/tcp
```

We can verify that this was successful using the `--list-ports` operation:

```
$ firewall-cmd --list-ports
```

output

```
5000/tcp
```

Individual Firewall Rules cont'd.

It is also possible to specify a sequential range of ports by separating the beginning and ending port in the range with a dash. For instance, if our application uses UDP ports 4990 to 4999, we could open these up on "public" by typing:

```
$ sudo firewall-cmd --zone=public --add-port=4990-4999/udp
```

After testing, we would likely want to add these to the permanent firewall. You can do that by typing:

```
$ sudo firewall-cmd --zone=public --permanent --add-port=5000/tcp
```

```
$ sudo firewall-cmd --zone=public --permanent --add-port=4990-  
4999/udp
```

```
$ sudo firewall-cmd --zone=public --permanent --list-ports
```

output

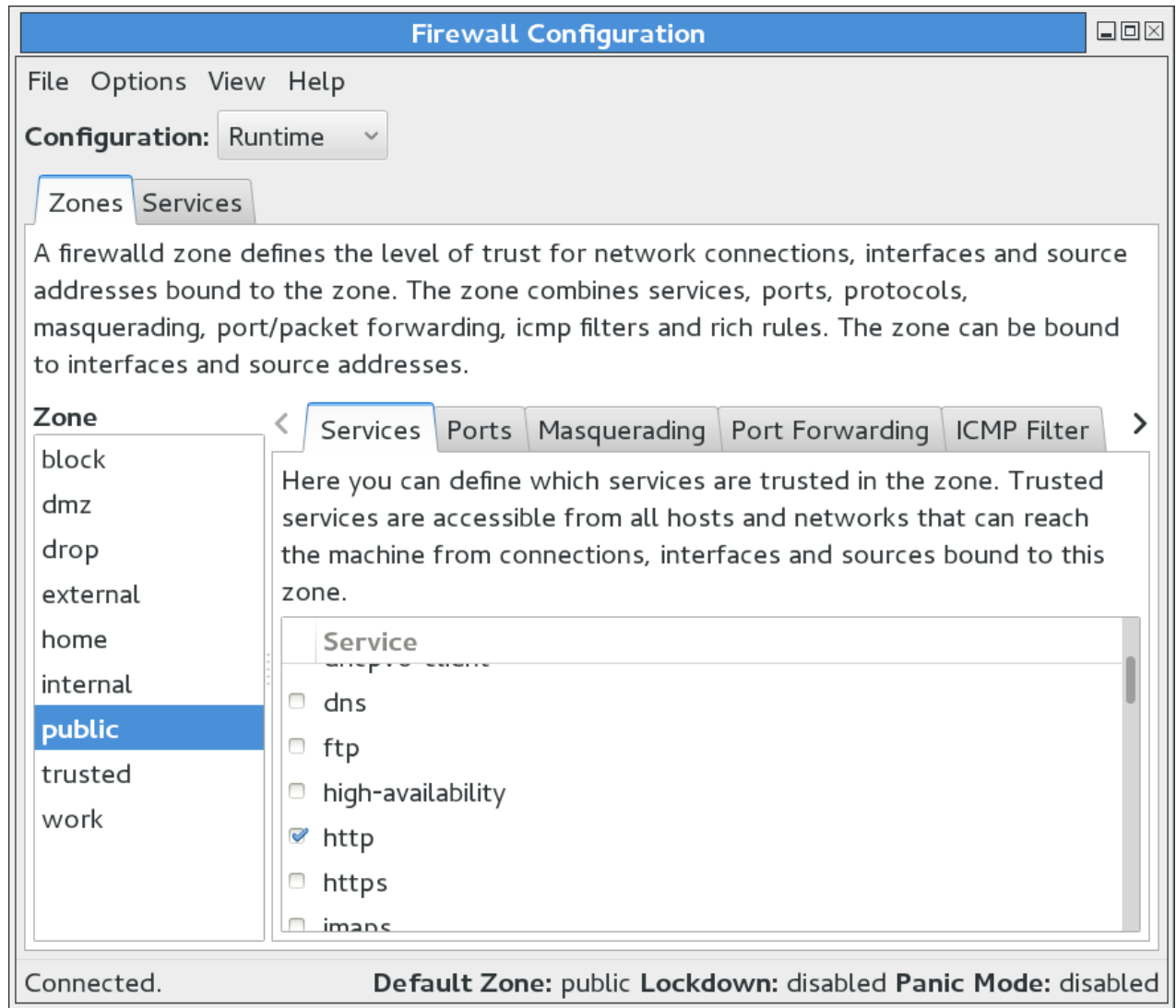
```
success
```

```
success
```

```
4990-4999/udp 5000/tcp
```

Firewalld GUI

The firewalld GUI is
invoked with:
`firewall-config`



Additional Firewall Functions

IP Masquerading

```
$ firewall-cmd --zone=public --query-masquerade
$ firewall-cmd --zone=public --add-masquerade
```

After enabling masquerading, you can set up port forwarding

```
$ firewall-cmd --zone=public --add-forward-port=port=22:proto=tcp:toport=3753
```

Or address forwarding

```
$ firewall-cmd --zone=external --add-forward-
port=port=22:proto=tcp:toaddr=192.0.2.55
```

Or both port & address forwarding:

```
$ firewall-cmd --zone=external /
--add-forward-port=port=22:proto=tcp:toport=2055:toaddr=192.0.2.55
```