RHEL7: How to get started with Firewalld.

certdepot.net/rhel7-get-started-firewalld

Share this link

0

0

Note: This is an RHCSA 7 exam objective and an RHCE 7 exam objective.

Presentation

Firewalld is the new userland interface in **RHEL 7**. It replaces the **iptables** interface and connects to the **netfilter** kernel code. It mainly improves the security rules management by allowing configuration changes without stopping the current connections.

To know if **Firewalld** is running, type:

```
# systemctl status firewalld
firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled)
   Active: active (running) since Tue 2014-06-17 11:14:49 CEST; 5 days ago
   ...
```

or alternatively:

```
# firewall-cmd --state
running
```

Note: If **Firewalld** is not running, the command displays **not running**.

If you've got several network interfaces in **IPv4**, you will have to activate **ip forwarding**. To do that, paste the following line into the **/etc/sysctl.conf** file:

```
net.ipv4.ip_forward=1
```

Then, activate the configuration:

```
# sysctl -p
```

Note: If you interested in kernel parameter configuration, there is a tutorial about the sysctl command.

Although **Firewalld** is the **RHEL 7** way to deal with firewalls and provides many improvements, iptables can still be used (but both shouldn't run at the same time).

You can also look at the iptables rules created by **Firewalld** with the **iptables-save** command.

Zone Management

Also, a new concept of zone appears: all network interfaces can be located in the same default zone or divided into different ones according to the levels of trust defined. In the latter case, this allows to restrict traffic based on origin zone (read this article from **lwn.net** for more details).
Note: Without any configuration, everything is done by default in the **public** zone. If you've got more than one network interface or use **sources** (see **Source management** section below), you will be able to restrict traffic between zones.

To get the default zone, type:

```
# firewall-cmd --get-default-zone
public
```

To get the list of zones where you've got network interfaces or sources assigned to, type:

```
# firewall-cmd --get-active-zones
public
  interfaces: eth0
```

Note: You can have more than one active zone at a time.

To get the list of all the available zones, type:

```
# firewall-cmd --get-zones
block dmz drop external home internal public trusted work
```

To change the default zone to **home** permanently, type:

```
# firewall-cmd --set-default-zone=home
success
```

Note: This information is stored in the **/etc/firewalld/firewalld.conf** file.

Network interfaces can be assigned to a zone in a **permanent** way.
To **permanently** assign the **eth0** network interface to the **internal** zone (a file called **internal.xml** is created in the **/etc/firewalld/zones** directory), type:

```
# firewall-cmd --permanent --zone=internal --change-interface=eth0
success
# nmcli con show | grep eth0
System eth0  4de55c95-2368-429b-be65-8f7b1a357e3f  802-3-ethernet  eth0
# nmcli con mod "System eth0" connection.zone internal
# nmcli con up "System eth0"
Connection successfully activated (D-Bus active path:
/org/freedesktop/NetworkManager/ActiveConnection/1)
```

Note1: This operation can also be done by editing the **/etc/sysconfig/network-scripts/ifcfg-eth0** file and add **ZONE=internal** followed by # **nmcli con reload** . It seems that with **RHEL 7.5**, the use of **ZONE** in **ifcfg-*** files no longer works (source).

Note2: More information about the **nmcli** command is available at the page dedicated to nmcli or at the IPV4 configuration page.
Note3: The **RHEL 7.3** release improves the way **Firewalld** handles zones (v0.3.9 -> v0.4.3.2: BZ#1302802).

To know which zone is associated with the **eth0** interface, type:

```
# firewall-cmd --get-zone-of-interface=eth0
internal
```

To get the **permanent** configuration of the **public** zone, type:

```
# firewall-cmd --permanent --zone=public --list-all
public (default, active)
  interfaces: eth0
  sources:
  services: dhcpv6-client ssh
  ports:
  masquerade: no
  forward-ports:
  icmp-blocks:
  rich rules:
```

It is also possible to create new zones. To create a new zone (here **test**), type:

```
# firewall-cmd --permanent --new-zone=test
success
# firewall-cmd --reload
success
```

Note: Only **permanent** zones can be created.

Source Management

A zone can be bound to a network interface (see above) and/or to a network addressing (called here a **source**).
Any network packet entering in the network stack is associated with a zone.
The association is done according to the following pattern:
– is the packet coming from a source already bound to a zone? (if yes, it is associated with this zone),
– if not, is the packet coming from a network interface already bound to a zone? (if yes, it is associated with this zone),
– if not, the packet is associated with the default zone.

This way, multiple zones can be defined even on a server with only one network interface!

Caution: To get this feature, **Firewalld** relies on **NetworkManager** (see reference). This means that if you plan to stop **NetworkManager** for any reason (for example when building a **KVM** host), you will have to stop Firewalld and use Iptables instead!

Note: With the **RHEL 7.3** release, **Firewalld** robustness has been improved in regard to **NetworkManager** (see details here).

To add a source (here **192.168.2.0/24**) to a zone (here **trusted**) **permanently**, type:

```
# firewall-cmd --permanent --zone=trusted --add-source=192.168.2.0/24
success
# firewall-cmd --reload
success
```

Note1: Use the **–remove-source** option to delete a previous assigned source.
Note2: Use the **–change-source** option to move the source to the new specified zone.
Note3: If you want to **temporarily** add a source to a zone, don't use the **–permanent** option and don't **reload** the firewall configuration. If you **reload** the firewall configuration, this will **cancel** all the operation.
Note4: You can also make some changes and when you like your new configuration, have it become your permanent configuration with the **firewall-cmd –runtime-to-permanent** command.

With the **RHEL 7.3** release, you can add a source based on a **MAC** address (here **00:11:22:33:44:55**) to a zone (here **trusted**) **permanently**:

```
# firewall-cmd --permanent --zone=trusted --add-source=00:11:22:33:44:55
success
# firewall-cmd --reload
success
```

With the **RHEL 7.3** release, you can create an **ipset** (a set of IP addresses or networks, see below) and add a source based on it:

```
# firewall-cmd --permanent --new-ipset=iplist --type=hash:ip
success
# firewall-cmd --reload
success
# firewall-cmd --ipset=iplist --add-entry=192.168.1.11
success
# firewall-cmd --ipset=iplist --add-entry=192.168.1.12
success
# firewall-cmd --permanent--zone=trusted --add-source=ipset:iplist
success
# firewall-cmd --reload
success
```

To get the list of the sources **currently** bound to a zone (here **trusted**), type:

```
# firewall-cmd --permanent --zone=trusted --list-sources
192.168.2.0/24 00:11:22:33:44:55 ipset:iplist
```

Note: Remove the **–permanent** option if you only want to display **temporary** settings.

To keep track of your configuration (**active** zones are zones that have a binding to an interface or source), type:

```
# firewall-cmd --get-active-zones
public
  interfaces: eth0
trusted
  sources: 192.168.2.0/24
```

As an exemple of source management, let’s assume you want to only allow connections to your server from a specific IP address (here **1.2.3.4/32**).

```
# firewall-cmd --zone=internal --add-service=ssh --permanent
success
# firewall-cmd --zone=internal --add-source=1.2.3.4/32 --permanent
success
# firewall-cmd --zone=public --remove-service=ssh --permanent
success
# firewall-cmd --reload
success
```

Source: Serverfault website.

With **RHEL 7.3**, a new option called **–info-zone** is available.
To get the detail of a zone called **public**, type:

```
# firewall-cmd --info-zone=public
public (active)
target: default
icmp-block-inversion: no
interfaces: eth0
sources:
services: dhcpv6-client ssh
ports:
protocols:
masquerade: no
forward-ports:
sourceports:
icmp-blocks:
rich rules:
```

Note: You can also add the **–permanent** option.

Service Management

After assigning each network interface to a zone, it is now possible to add services to each zone.
To allow the **http** service **permanently** in the **internal** zone, type:

```
# firewall-cmd --permanent --zone=internal --add-service=http
success
# firewall-cmd --reload
success
```

Note1: Type **–remove-service=http** to deny the **http** service.
Note2: The **firewall-cmd –reload** command is necessary to activate the change. Contrary to the **–complete-reload** option, current connections are not stopped.
Note3: If you only want to **temporarily** add a service, don't use the **–permanent** option and don't **reload** the firewall configuration. If you **reload** the firewall configuration, you **cancel** all the operation.

If you want to temporary add several services (here **http**, **https**, and **dns**) at the same time in the **internal** zone, type:

```
# firewall-cmd --zone=internal --add-service={http,https,dns}
success
```

To get the list of services in the default zone, type:

```
# firewall-cmd --list-services
dhcpv6-client ssh
```

Note: To get the list of the services in a particular zone, add the **–zone=** option.

With **RHEL 7.3**, a new option called **–info-service** is available.
To get some information about the **ftp** service, type:

```
# firewall-cmd --info-service=ftp
ftp
  ports: 21/tcp
  protocols:
  source-ports:
  modules: nf_conntrack_ftp
  destination:
```

Note: You can also add the **–permanent** option.

Firewall Services Configuration

With the **Firewalld** package, the firewall configuration of the main services (ftp, httpd, etc) comes in the **/usr/lib/firewalld/services** directory. But it is still possible to add new ones in the **/etc/firewalld/services** directory. Also, if files exist at both locations for the same service, the file in the **/etc/firewalld/services** directory takes precedence.

For example, it is the case of the **HAProxy** service. There is no firewall configuration associated.
Create the **/etc/firewalld/services/haproxy.xml** and paste the following lines:

```
<?xml version="1.0" encoding="utf-8"?>
<service>
 <short>HAProxy</short>
 <description>HAProxy load-balancer</description>
 <port protocol="tcp" port="80"/>
</service>
```

Note: You can use the **firewall-cmd –permanent –new-service=haproxy** command to quickly create a configuration file skeleton.

Assign the correct **SELinux** context and file permissions to the **haproxy.xml** file:

```
# cd /etc/firewalld/services
# restorecon haproxy.xml
# chmod 640 haproxy.xml
```

Add the **HAProxy** service to the default zone **permanently** and **reload** the firewall configuration:

```
# firewall-cmd --permanent --add-service=haproxy
success
# firewall-cmd --reload
success
```

Note: According to **Bert Van Vreckem**, it is possible to go quicker by using the command history (see details here):

```
# firewall-cmd --add-service=haproxy
success
# firewall-cmd --add-service=haproxy --permanent
success
```

In **RHEL 7.0** (**Firewalld v0.3.9.7)**, there were **47** firewall services configured: **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**, **mountd**, **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**.
In **RHEL 7.1** (**Firewalld v0.3.9.11)**, the **RH-Satellite-6** service was added.
In **RHEL 7.2** (**Firewalld v0.3.9.14**), the **freeipa-ldaps**, **freeipa-ldap**, **freeipa-replication**, **iscsi-target**, **rsyncd** and **vdsm** services were added.
In **RHEL 7.3** (**Firewalld v0.4.3.2**), the **amanda-k5-client**, **ceph**, **ceph-mon**, **docker-registry**, **dropbox-lansync**, **imap**, **kadmin**, **mosh**, **pop3**, **privoxy**, **ptp**, **pulseaudio**, **puppetmaster**, **sane**, **smtps**, **snmp**, **snmptrap**, **squid**, **synergy**, **syslog**, **syslog-tls**, **tinc**, **tor-socks**, **xmpp-bosh**, **xmpp-client**, **xmpp-local** and **xmpp-server** services have been added for a total of **81** services.

Port Management

Port management follows the same model as service management.

To allow the **443/tcp** port **temporarily** in the **internal** zone, type:

```
# firewall-cmd --zone=internal --add-port=443/tcp
success
```

Note1: To make the configuration **permanent**, add the **–permanent** option and **reload** the firewall configuration.
Note2: Type **–remove-port=443/tcp** to deny the port.

To get the list of ports **currently** open in the **internal** zone, type:

```
# firewall-cmd --zone=internal --list-ports
443/tcp
```

Note: To only get the list of ports **permanently** open, add the **–permanent** option. Here, you will not get anything.

Rich Rules

As the syntax used by the rich rules are somehow difficult to remember, keep in mind the **man firewalld.richlanguage** command and the **Example** section at the end.

Here is the format of a rich rule:

```
# firewall-cmd --add-rich-rule 'rule ...'
```

To allow all connections from **192.168.2.2**, type:

```
# firewall-cmd --add-rich-rule 'rule family="ipv4" source address="192.168.2.2" log
accept'
```

Note1: The **log** option writes coming packets into the **/var/log/messages** file.
Note2: Use the **–remove-rich-rule** option instead of the **–add-rich-rule** option if you want to delete an already existing rule.

To list the rich rules set in the default zone, type:

```
# firewall-cmd --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: eth0
  sources:
  services: dhcpv6-client ssh
  ports:
  protocols:
  masquerade: no
  forward-ports:
  sourceports:
  icmp-blocks:
  rich rules:
        rule family="ipv4" source address="192.168.2.2" log accept
```

Direct Rules

It is still possible to set specific rules by using the **direct** mode (here to open the tcp port **9000**) that by-passes the **Firewalld** interface:

```
# firewall-cmd --direct --add-rule ipv4 filter INPUT 0 -p tcp --dport 9000 -j ACCEPT
success
```

Note1: This example has been borrowed from Khosro Taraghi's blog.
Note2: Use the same command with the **–remove-rule** instead of **–add-rule** to delete the rule.
Note3: The configuration is **temporary** except if you add the **–permanent** option just after the **–direct** option.
Note4: It is not necessary to **reload** the firewall configuration, all commands are **directly** activated.

To display all the direct rules added, type:

```
# firewall-cmd --direct --get-all-rules
```

Note1: For information, the configuration is written into the **/etc/firewalld/direct.xml** file.
Note2: **Direct rules** are not part of the **RHCSA**/**RHCE** exam objectives.

IP Set Management

With the **RHEL 7.3** comes the ability to create **ipsets**. An **ipset** is a set of IP addresses or networks. The different categories belong to **hash:ip** or **hash:net**.

To create a permanent IPv4 **ipset** containing two IP addresses and drop packets coming from these addresses, type:

```
# firewall-cmd --permanent --new-ipset=blacklist --type=hash:ip
success
# firewall-cmd --reload
success
# firewall-cmd --ipset=blacklist --add-entry=192.168.1.11
success
# firewall-cmd --ipset=blacklist --add-entry=192.168.1.12
success
# firewall-cmd --add-rich-rule='rule source ipset=blacklist drop'
success
```

Note: Add **–option=family=inet6** to create an **IPv6** ipset.

To get the content of the **blacklist** ipset, type:

```
# firewall-cmd --info-ipset=blacklist
blacklist
type: hash:ip
options:
entries: 192.168.1.11 192.168.1.12
```

To remove the **192.168.1.12** entry from the **blacklist** ipset, type:

```
# firewall-cmd --ipset=blacklist --remove-entry=192.168.1.12
success
# firewall-cmd --ipset=blacklist --get-entries
192.168.1.11
```

To create a permanent IPv4 ipset containing two networks, type:

```
# firewall-cmd --permanent --new-ipset=netlist
success
# firewall-cmd --reload
success
# firewall-cmd --ipset=netlist --add-entry=192.168.1.0/24
success
# firewall-cmd --ipset=netlist --add-entry=192.168.2.0/24
success
# firewall-cmd --info-ipset=netlist
netlist
 type: hash:net
 options:
 entries: 192.168.1.0/24 192.168.2.0/24
```

To remove the **netlist** ipset, type:

```
# firewall-cmd --permanent --delete-ipset=netlist
success
# firewall-cmd --reload
success
# firewall-cmd --get-ipsets
blacklist
```

It is also possible to download the content of an ipset from a file (**--add-entries-from-file=file** option) or store it with the name **ipset** in the **/etc/firewalld/ipsets/ipset.xml** or **/usr/lib/firewalld/ipsets/ipset.xml** files according to the following format:

```
<?xml version="1.0" encoding="utf-8"?>
<ipset type="hash:ip">
  <short>My Ipset</short>
  <description>description</description>
  <entry>192.168.1.11</entry>
  <entry>192.168.1.12</entry>
</ipset>
```

To load this ipset, type:

```
# firewall-cmd --reload
```

Masquerading

If your firewall is your network gateway and you don’t want everybody to know your internal addresses, you can set up two zones, one called **internal**, the other **external**, and configure **masquerading** on the **external** zone. This way, all packets will get your firewall ip address as source address.

To set up **masquerading** on the **external** zone in a temporary way, type:

```
# firewall-cmd --zone=external --add-masquerade
success
```

Note1: To remove **masquerading**, use the **–remove-masquerade** option.
Note2: To know if **masquerading** is active in a zone, use the **–query-masquerade** option.
Note3: To get the configuration **permanent**, add the **–permanent** option and **reload** the firewall configuration.

Port Forwarding

Port forwarding is a way to forward inbound network traffic for a specific port to another internal address or an alternative port.

Caution: Port forwarding requires masquerading (source). This point is a classical mistake made during the **RHCE** exam.

So, you need to enable **masquerading** before anything else:

```
# firewall-cmd --zone=external --add-masquerade
success
```

If you want all packets intended for port **22** to be now forwarded to port **tcp 3753 temporarily**, type:

```
# firewall-cmd --zone=external --add-forward-port=port=22:proto=tcp:toport=3753
success
```

Note1: To remove **port forwarding**, use the **–remove-forward-port** option.
Note2: To know if **port forwarding** is active in a zone, use the **–query-forward-port** option.
Note3: If you want to make the configuration **permanent**, add the **–permanent** option and reload the firewall configuration.

Also, if you want to define the destination ip address, this time in a **permanent** way, type:

```
# firewall-cmd --permanent --zone=external --add-forward-
port=port=22:proto=tcp:toport=3753:toaddr=10.0.0.1
success
# firewall-cmd --reload
success
```

Special Modules

Sometimes it is required to download specific modules. Instead of using a rc.local file, it is better to notify **Firewalld** through the **/etc/modules-load.d** directory.
In this example we want to add the **ip_nat_ftp** and **ip_conntrack_ftp** modules to follow **ftp** connections.
We only need to choose a filename (here **firewall_ftp.conf**) and type these instructions:

```
# echo ip_nat_ftp > /etc/modules-load.d/firewall_ftp.conf
# echo ip_conntrack_ftp >> /etc/modules-load.d/firewall_ftp.conf
```

Source: StackExchange website.

Offline Configuration

In some cases (installations through **Anaconda** or **Kickstart** for example), you need to set up firewall rules when **Firewalld** is not running. The **firewall-offline-cmd** command has just been created for this purpose.
For instance, to open the **tcp port 22**, you would type in the **/etc/sysconfig/iptables** file:

```
-A INPUT -p tcp -m state --state NEW -m tcp --dport 22 -j ACCEPT
```

Instead, you can now execute the following command:

```
# firewall-offline-cmd --direct --add-rule ipv4 filter INPUT 0 -p tcp -m state --state
NEW -m tcp --dport 22 -j ACCEPT
```

Configuration Backup

To store the current configuration into files, type:

```
# iptables -S > firewalld_rules_ipv4
# ip6tables -S > firewalld_rules_ipv6
```

Debugging Tips

To better understand how **Firewalld** works, assign the **'–debug'** value to the **FIREWALLD_ARGS** variable in the **/etc/sysconfig/firewalld** file:

```
# firewalld command line args
# possile values: --debug
FIREWALLD_ARGS='--debug'
```

Restart the **Firewalld** daemon:

```
# systemctl restart firewalld
```

Note: Messages will be written into the **/var/log/firewalld** file.

Also, with the **RHEL 7.3** release comes the **LogDenied** directive in the **/etc/firewalld/firewalld.conf** file.
This directive adds logging rules right before reject and drop rules in the **INPUT**, **FORWARD** and **OUTPUT** chains for the default rules and also final reject and drop rules in zones.
Possible values are: **all**, **unicast**, **broadcast**, **multicast** and **off** (value by default).

Reload the **Firewalld** configuration:

```
# firewall-cmd --reload
```

Note: Messages will be written into the **/var/log/messages** file. If you also want messages to be written in a file called **/var/log/custom.log**, edit the **/etc/rsyslog.conf** file, add the line **kern.warning /var/log/custom.log** and restart the **rsyslog** configuration with **# systemctl restart rsyslog**

Additional Resources

In addition, you can:

- read this article about Firewalld by Sander van Vugt,
- watch **Thomas Woerner**'s video about Firewalld, present and future (48min/2015),
- read this FedoraProject page about Fail2ban with Firewalld,
- read this article about Firewalld and zone deployment by James Hogarth ,
- read the CIS RHEL 7 Server Hardening Guide,
- watch **Venkat Nagappan**'s video about Firewalld Concepts and Examples (34min/2015),
- watch **Sander van Vugt**'s video about port forwarding using firewall-cmd (8min/2015),
- watch **Ralph Nyberg**'s video about Firewalld and Iptables (26min/2016),
- read **Thomas Woerner**'s blog, the blog of the author of Firewalld,
- read this presentation from the 11th Netfilter Workshop (2015),
- read the changelog of the Firewalld versions,
- have a look at **Daniel Aleksandersen**'s blog about Configuring zones bound by source IPs in Firewalld or Comparing and contrasting Uncomplicated Firewall and Firewalld,
- read **Alexander Molochko**'s blog about Creating a new zone in Firewalld.

Sources: RHEL7 Security Guide, wiki Fedora project.

Test yourself!