Chapter 5:
Configuring a firewall

Learning objectives

Upon completing this chapter, the learner should be able to:

- Set up a basic firewall environment
- Manage firewalld services
- Manage firewalld zones
- Allow access to specific services on your computer
- Set up a base firewall by using the firewall cmd command-line tool

Key terms

firewall

Zone

iptables

firewalld

Block zone

Dmz zone

Drop zone

External zone

Home zone

Internal zone

Public zone

Trusted zone

workfirewalld service

firewall-cmd

firewalld rich rule

Table of content

1. Understanding Linux firewalling

A firewall is a protective layer that is configured between a private and a public network to segregate traffic. There are several types of firewalls, one of which performs data packet filtering. A data packet is formed as a result of a process called encapsulation whereby the header information is attached to a message during the formation of the packet. The header includes information such as source and destination IP addresses, port, and type of data.

Based on pre-defined rules, a firewall intercepts each inbound and outbound data packet, inspects its header, and decides whether to allow the packet to pass through or not. A port is defined in the /etc/services file for each network service available on the system, and is typically standardized across all network operating systems, including RHEL. Some common services and the ports they listen on are: ftp on port 21, ssh on 22, telnet on 23, postfix on 25, http on 80, and ntp on port 123.

```
[root@server1:~]# vim /etc/services
...
ftp-data        20/tcp
ftp-data        20/udp
 21 #is registered to ftp, but also used by fsp
ftp             21/tcp
ftp             21/udp          fsp fspd
ssh               22/tcp                          # The
Secure Shell (SSH) Protocol
ssh               22/udp                          # The
Secure Shell (SSH) Protocol
telnet          23/tcp
telnet          23/udp
 - 24 #private mail system
lmtp              24/udp                        # LMTP
Mail Delivery
smtp            25/tcp          mail
smtp            25/udp          mail
time            37/tcp          timserver
time            37/udp          timserver
```

Netfilter allows kernel modules to inspect every incoming, outgoing, or forwarded packet and act upon such a packet by either allowing it or blocking it. So, the kernel firewall allows for inspection of incoming packets, outgoing packets, and packets that are traversing from one interface to another if the RHEL server is providing routing functionality.

1.1. Understanding previous solutions

To interact with netfilter, different solutions can be used. On earlier versions of Red Hat Enterprise Linux, iptables was the default solution to configure netfilter packet filtering. This solution worked with the command-line utility iptables, which provided a sophisticated and detailed way of defining firewall rules, but that was also challenging to use for the occasional administrator because of the complicated syntax of iptables commands and because the ordering rules could become relatively complex. The iptables service is still offered on Red Hat Enterprise Linux7. It is not recommended as the default service, though, and it cannot be used on a server where firewalld is used as well.

When working with firewalld, you should no longer use iptables and related services. That is because these services are incompatible to one another, and making changes to the iptables configuration will affect firewalld as well, so they must be avoided. On a server where multiple administrators are working, you risk that a less-knowledgeable administrator wants to create a firewall configuration and notices that the iptables service is not running and wants to start the iptables service anyway. This might mess up your firewalld-based firewall configuration.

Systemd provides a nice solution to make sure that unwanted services are not started by accident: You can use systemctl mask to exclude them from ever being started.

To exclude all iptables-based services from ever being started, type in the following command, which creates a symbolic link to /dev/null for the related service files in /etc/systemd/system.

```
[root@server1  ~] # for i in iptables ip6tables ebtables;
do systemctl mask $i; done
ln -s '/dev/null' '/etc/systemd/system/iptables.service'
ln -s '/dev/null' '/etc/systemd/system/ip6tables.service'
```

Notice that this is an elegant way to disable services. Service files in /etc/systemd/system always take precedence over the configuration files in /usr/lib/systemd/system. By linking the iptables-related service scripts in /etc/systemd/system to /dev/null, they will never start, but it is easy to enable them again by just removing these symbolic links or by using the command: for i in iptables ip6tables ebtables; do systemctl unmask $i; done.

1.2. Understanding firewalld

In Red Hat Enterprise Linux7 a new method was introduced: Firewalld is a system service that can configure firewall rules by using different interfaces. Administrators can manage rules in a firewalld environment, but even more important is that applications can request ports to be opened using the DBus messaging system, which means that rules can be added or removed without any direct action required of the system administrator.

Firewalld was developed as a complete new solution for managing Linux firewalls. It uses the firewalld service to manage the netfilter firewall configuration, and it is incompatible with the iptables service, so do not ever use firewalld and iptables on the same system; they are mutually exclusive.

1.3. Understanding firewalld zones

Firewalld makes firewall management easier by working with zones. A zone is a collection of rules that are applied to incoming packets matching a specific source address or network interface. Firewalld applies to incoming packets only by default, and no filtering is happening on outgoing packets.

The use of zones is particularly important on servers that have multiple interfaces, where zones allow administrators to easily assign a specific set of rules. On servers that have just one network interface, you might do very well with just one zone, which is the default zone. In general, every packet that comes into a system is analyzed for its source address, and based on that source address, firewalld analyzes to see whether the packet belongs to a specific zone. If that is not the case, the zone for the incoming network interface is used. If no specific zone is available, the packet is handled by the settings in the default zone.

Firewalld works with some default zones, the following table describes these default zones.

Zone name	Default Settings
Block	Incoming network connections are rejected with an “icmp–host–prohibited” message. Only network connections that were initiated on this system are allowed.
Dmz	For use on computers in the demilitarized zone. Only selected incoming connections are accepted, and limited access to the internal network is allowed.
Drop	Any incoming packets are dropped and there is no reply
External	For use on external networks with masquerading (Network Address Translation [NAT]) enabled, used especially on routers. Only selected incoming connections are accepted.
Home	For use with home networks. Most computers on the same network are trusted, and only selected incoming connections are accepted.
Internal	For use in internal networks. Most computers on the same network are trusted, and only selected incoming connections are accepted.
Public	For use in public areas. Other computers in the same network are not trusted, and limited connections are accepted. This is the default zone for all newly created network interfaces.
trusted	All network connections are accepted.
work	For use in work areas. Most computers on the same network are trusted, and only selected incoming connections are accepted.

1.4. Understanding firewalld configuration Files

The second key element while working with firewalld is the service. Note that a service in firewalld is not the same as a service in systemd. In firewalld, some default services are defined, which allows administrators to easily allow or deny access to specific ports on a server.

Behind each service is a configuration file that explains which UDP or TCP ports are involved, and if so required, which kernel modules must be loaded.

The firewalld service stores firewall rules in XML file format at two different locations: the system-defined rules in the /usr/lib/firewalld directory and the user-defined in /etc/firewalld. The files at the former location can be used as templates for adding or modifying new rules. We simply need to copy the required file to the /etc/firewalld/services directory and make necessary updates.

The firewalld service reads the files located in /etc/firewalld and applies the rules defined in them. A listing of both directories is shown below:

```
[root@server1:~]# ll /etc/firewalld/
total 8
-rw-r--r--. 1 root root 2006 Oct 18 19:02 firewalld.conf
drwxr-x---. 2 root root    6 Oct 18 19:02 helpers
drwxr-x---. 2 root root    6 Oct 18 19:02 icmptypes
drwxr-x---. 2 root root    6 Oct 18 19:02 ipsets
-rw-r--r--. 1 root root  272 Oct 18 19:02 lockdown-
whitelist.xml
drwxr-x---. 2 root root    6 Oct 18 19:02 services
drwxr-x---. 2 root root   46 Oct 18 19:02 zones
[root@server1:~]# ll /usr/lib/firewalld/
total 16
drwxr-xr-x. 2 root root  224 Dec  2 11:20 helpers
drwxr-xr-x. 2 root root 4096 Dec  2 11:20 icmptypes
drwxr-xr-x. 2 root root   20 Dec  2 11:20 ipsets
drwxr-xr-x. 2 root root 8192 Dec  2 11:20 services
drwxr-xr-x. 2 root root  163 Dec  2 11:20 zones
[root@server1:~]# ls /usr/lib/firewalld/services/
amanda-client.xml                              dhcp.xml
https.xml          ldap.xml                 nmea-0183.xml
puppetmaster.xml     snmptrap.xml             vdsm.xml
...
dhcpv6.xml                           high-availability.xml
ldaps.xml          nfs.xml                  pulseaudio.xml
smtp.xml             upnp-client.xml
```

A Listing shows what the contents of the ftp service file looks like.

```
[root@server1 services]# cat /usr/lib/firewalld/services/ftp.xml
<?xml version="1.0" encoding="utf-8"?>
<service>
<short>FTP</short>
<description>FTP is a protocol used for remote file transfer. If you
plan to make your FTP
server publicly available, enable this option. You need the vsftpd
package installed for this
option to be useful.</description>
<port protocol="tcp" port="21"/>
<module name="nf_conntrack_ftp"/>
</service>
```

As we can see, the file has a short and long description for the service, and it also tells us the protocol and port number that it uses.

2. Working with firewall-cmd

To configure a firewall with firewalld, two tools are available for this purpose:

- the command-line tool firewall-cmd
- and the firewall-config tool, which has a graphical interface

It is a good idea to focus your efforts on the firewall-cmd tool, this easily accessible tool enables uncomplicated configuration. If you prefer working with the graphical firewall-config tool, that is possible, too. You can do everything with the graphical tool that you can do with the command-line tool.

When working with either of these tools, be aware of where exactly modifications are made. Both tools work with an in-memory state of the configuration in addition to an on-disk state (permanent state) of the configuration. While using either of these tools, make sure to commit changes to disk before proceeding.

First thing to know is how to list the available services in the system, the listing use firewall-cmd –get --services for a list of all available services:

```
[root@server1 ~] # firewall-cmd --get-services
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
```

The firewall-cmd interface offers many more options, the following table describes some of the most important command line options.

Firewall-cmd Options	Explanation
--get-zones	Lists all available zones
--get-default-zone	Shows the zone currently set as default zone
--set-default-zone=<ZONE>	Changes the default zone
--get-services	Shows all available services
--list-services	Shows services currently in use
--add-service=<service-name> [--zone=<ZONE>]	Adds a service to the current default zone or the zone that is specified
--remove-service=<service-name>	Removes a service from the configuration
--list-all [--zone=<ZONE>]	Lists all configurations in a zone
--add-port=<port/protocol> [--zone=<ZONE>]	Adds a port and protocol
--remove-port=<port/protocol> [--zone=<ZONE>]	Removes a port from the configuration
--add-interface=<INTERFACE> [--zone=<ZONE>]	Adds an interface to the default zone or a specific zone that is specified
--remove-interface=<INTERFACE> [--zone=<ZONE>]	Removes an interface from a specific zone
--add-source=<ipaddress/netmask> [--zone=<ZONE>]	Adds a specific IP address
--remove-source=<ipaddress/ Netmask> [--zone=<ZONE>]	Removes an IP address from the configuration
--permanent	Writes configuration to disk and not to run-time
--reload	Reloads the on-disk configuration

2.1. Managing firewall rules

The firewall-cmd command is a powerful command line tool that is used to create, modify, manage, and remove rules for the firewalld service. This tool is an easily accessible tool that enables administrators to change the runtime configuration of the firewall and to write this configuration to disk.

In the following exercise you will display the name of the default zone and then add three rules to it. You will first add and activate a permanent rule to allow HTTP traffic on port 80, and then add a runtime rule for traffic intended for TCP port 443 (the HTTPS service). You will add another permanent rule to allow VNC traffic on any TCP port between 5901 and 5910. You will run appropriate commands to confirm your changes, and also display the contents of the default zone file to see the addition of the permanent rule.

Exercise: Add and manage firewall rules

1. Determine the name of the current default zone:

   #firewall-cmd --get-default-zone

   public

2. Add a permanent rule to allow HTTP traffic on its default port:

   #firewall-cmd --permanent --add-service=http

   success

   This command has added the http service name to the default zone file public.xml located in the /etc/firewalld/zones/ directory. When this new rule is activated, it will use the port and protocol defined in the template file /usr/lib/firewalld/services/http.xml.

3. Activate the new rule:

   #firewall-cmd --reload

   success

4. Add a runtime rule to allow traffic on TCP port 443:

   #firewall-cmd --add-port=443/tcp

5. Add a permanent rule for VNC traffic on TCP port range 5901 to 5910, and activate it immediately:

   #firewall-cmd --permanent --add-port=5901-5910/tcp; firewall-cmd --reload

6. List the services and ports to confirm the success of all three changes:

   #firewall-cmd --list-services

   dhcpv6-client http ssh

   #firewall-cmd --list-ports

   443/tcp 5901-5910/tcp

7. Display the contents of the default zone file to confirm the addition of both permanent rules, and notice that the firewall-cmd command makes a copy of the affected zone file with .old extension whenever an update is made to the zone.

In this exercise, you will modify the rules that you added in the previous exercise and then delete two of them. You will remove the permanent rules for both HTTP (port 80) and HTTPS (port 443), and restrict the port range for VNC service to 5901 and 5902. You will run appropriate commands to confirm the changes, and also display the contents of the default zone file to validate the modifications.

Exercise: Modify and remove firewall rules

1. Remove the permanent rule for HTTP:

   #firewall-cmd --permanent --remove-service=http

2. Remove the runtime rule for port 443, and confirm the immediate deletion:

   #firewall-cmd --remove-port=443/tcp

   #firewall-cmd –list-ports

   5901-5910/tcp

   Since it was a runtime rule, running the first command above deleted it right away.

3. Remove the permanent rule for ports 5901 to 5910, and add a new permanent rule to restrict VNC to listen only on 5901 and 5902 ports:

   #firewall-cmd --permanent --remove-port=5901-5910/tcp

   #firewall-cmd --permanent --add-port=5901-5902/tcp

4. Activate the new permanent changes:

   #firewall-cmd --reload

5. List the services and ports to confirm the success of all three changes:

   #firewall-cmd --list-services

   dhcpv6-client ssh

   #firewall-cmd --list-ports

   5901-5902/tcp

   Both outputs reflect the updates.

2.2. Creating firewall services

In some cases, you need to add ports to the firewall configuration that are not included by default. If that is the case, you can use the -- add -port option with the firewall-cmd command, but you can create a service configuration file as an alternative, this can be done by editing the XML code. As an alternative, you can use firewall-cmd –new -service to do this. See man firewall-cmd for more information In the following exercise you will learn how to create a custom service file that allows the ssh service to be accessed on port 2022.

Exercise: Creating a custom firewalld service

1. Because it is much easier to base new service files on existing service files, use the command cp /usr/lib/firewalld/services/ssh.xml /etc/firewalld/services/ssh-custom.xml.
2. Open the file /etc/firewalld/services/ssh with an editor.
3. In the service file, change the port that is set to 22 to a new port (2022).
4. Modify the description setting to show that this is a modified service file, and save changes to disk.

5. Type firewall-cmd –get -services. Notice that you do not see the new service file listed yet.
6. Type firewall-cmd --reload and repeat firewall-cmd –get -services. You'll see the new service file listed now.
7. Type firewall-cmd –add -service ssh -custom --permanent, followed by firewall-cmd --reload.
8. Type firewall-cmd –list -services. You'll see the new service file added to the default zone.
9. Notice that this exercise has shown how to create a firewall service. To make the process available on the new port, modify the process configuration also.

3. Configuring firewalld rich rules

Up to now, you have worked with firewalld services. Although convenient, the options that are offered by firewalld services are sometimes a bit limited. That is why firewalld offers alternative solutions for allowing traffic as well. Currently, there are two solutions:

- Direct rules allow administrators to add detailed hand code rules into the firewall configuration. They offer advanced features and are somewhat hard to manage. They use a syntax that looks a lot like iptables syntax and allow administrators to work with firewalld in an iptables-like way.
- Rich rules give administrators an easy-to-use expressive language to define custom rules that cannot be created using the default firewalld syntax.

3.1. Rich rule syntax

Rich rules are used to create allow/deny rules, but with advanced options, such as the following:

- Logging configuration
- Port forwarding
- Masquerading
- Rate limiting
- Allow/deny connections for one specific zone

The basic syntax of a rich rule is as follows:

- Rule
- [source] [destination]
- {service|port|protocol|icmp-block|masquerade|forward-port}
- [log] [audit]
- [accept|reject|drop]

3.2. Rich rules ordering

When working with rich rules, it is easy to create conflicting rules. For instance, you may deny access to an entire network but want to allow access to one specific node in that network. Because of this, ordering becomes important as well when working with rich rules. The basic ordering rules within zones are as follows:

1. Direct rules
2. Port forwarding and masquerading rules
3. Logging rules
4. Allow rules
5. Deny rules

Typically, a rule that will not be matched by anything will be denied, but that depends on the default zone configuration as well. If the trusted zone is used, for instance, packets that are not matching anything are allowed.

3.3. Managing Rich Rules

There are four basic manipulations when working with rich rules, the table below describes these manipulations.

Manipulation	Explanation
--add-rich-rule='<RULE>'	Adds <RULE> to the default zone or to the zone that is specified.
--remove-rich-rule='<RULE>'	Removes <RULE> from the default zone or from the zone that is specified.
--query-rich-rule='<RULE>'	Queries if <RULE> has been added to a zone. Returns 0 if the rule is present and 1 if it is not and does not give any further details.
--list-rich-rules	Lists all rich rules for the default zone or for the zone that is specified as an argument.

To make testing of rich rules easier, you can add rich rules to the runtime configuration with a timeout. Once the timeout has passed, the rich rule is automatically removed. This ensures that you will not be locked out after making an error to the configuration of rich rules, which is useful when configuring a firewall on a remote server. To add a timeout to a rich rule, add the --timeout=XX to the end of the firewall-cmd rule.

To help you learn the rich rule syntax, Exercise walks you through the procedure to add some rich rules to a zone. Notice that none of the rules is set to --permanent, because this exercise is to demonstrate rich rule workings only.

Exercise: Using rich rules

1. From a root shell, type firewall-cmd –zone=dmz –add –rich –rule='rule family=ipv4 source address=10.0.0.100/32 reject'–timeout=60.
2. Type firewall-cmd --list-all –zone=dmz to verify that the rule has been added successfully.
3. Now enter firewall-cmd –add –rich –rule='rule service name=http log limit value=3/m accept' –zone=dmz.
4. Type firewall-cmd –list -all –zone=dmz again to verify that the new rule was added successfully. Also notice that the rule you have added in step 1 of this exercise is now gone.
5. Type firewall-cmd –add –rich –rule='rule protocol value=icmp accept' –zone=dmz. This rule allows all Internet Control Message Protocol (ICMP) traffic toward the demilitarized zone (DMZ).
6. Type firewall-cmd –add –rich –rule='rule family=ipv4 source address=10.0.0.0/24 port port=7900 -7905 protocol=tcp accept' –zone=dmz.
7. Verify that all rich rules have successfully been added, by using firewall-cmd –list -all –zone=dmz.

Quiz

Chapter review questions

1. Which of the following is not a standard firewalld zone?
   a. Untrusted
   b. Trusted
   c. External
   d. Internal
2. Which of the following is the name of firewalling as implemented in the Linux kernel?
   a. iptables
   b. firewalld
   c. netfilter
   d. firewall – mod
3. Which of the following is not an advantage of firewalld?
   a. Rules can be modified through dbus
   b. It has an easy to use command – line interface
   c. It has an easy to use graphical interface
   d. It can be used as an enhancement to iptables
4. Which command enables you to list all available firewalld services?
   a. firewall-cmd --list-services
   b. firewall-cmd --list-all
   c. firewall-cmd --get-services
   d. firewall-cmd --show-services

5. What is the name of the GUI tool that enables you to easily manage firewalld configurations?
   a. System – config – firewall
   b. Firewall – gtk
   c. Firewall – config
   d. Firewall – gui
6. What is the name of the CLI tool that enables you to easily manage firewalld configurations?
   a. System – config – firewall
   b. Firewall – cmd
   c. Firewall – config
   d. Firewall – gui
7. Which zones should you use for an interface that is on a network where you need minimal firewall protection because every other computer on that same network is trusted?
   a. Trusted
   b. Home
   c. external
   d. Private

8. Which of the following statements is true about the --permanent command line option when used with firewall-cmd?
   a. Configuration that is added using --permanent is activated immediately and will be activated automatically after (re)starting firewalld.
   b. Configuration that is added using --permanent is activated immediately.
   c. Configuration that is added using --permanent is not activated immediately and can be activated only by using systemctl restart firewalld.
   d. To activate configuration that has been added with the –permanent option, you need to reload the firewall configuration by using firewall-cmd –reload.
9. Which command enables you to get an overview of all the current firewall configurations for all zones?
   a. Firewall - cmd –show - current
   b. Firewall - cmd –list - all
   c. Firewall - cmd –list - current
   d. Firewall - cmd –show - all
10. Iptables is a packet-filtering firewall.
   a. True
   b. False
11. When the firewalld service is used for managing the firewall, a couple of services should never be running on your server. What services should not be running when you are using firewalld?
   a. iptables
   b. ebtables
   c. ip6tables
   d. network

12. In a firewalld configuration, you can use different building blocks. These building blocks are processed in a specific order, and are shown in the following list. Which answer lists their correct order?

- Deny rules
- Logging rules
- Direct rules
- Allow rules
- Port forwarding and masquerading rules

a. 3, 5, 2, 4, 1

b. 5, 3, 2, 4, 1

c. 3, 5, 4, 1, 2

d. 2, 1, 5, 3, 4

13. Which of the following is the typical location for custom firewalld service files?

a. /etc/systemd/system

b. /usr/lib/firewalld/services

c. /etc/firewalld/services

d. /etc/firewalld

14. Which of the following cannot be configured using firewalld rich rules?

a. Logging

b. Filtering based on one specific IP address instead of all IP addresses assigned to a zone

c. Custom port allocations

d. Rate limiting

15. The basic syntax of a rich rule is as follows:

Rule – [source] [destination] – {service|port|protocol|icmp-block|masquerade|forward-port} [log] [audit] [accept|reject|drop]

a. True

b. False

16. Lists all rich rules for the default zone or for the zone that is specified as an argument.

a. --query-rich-rule='<RULE>'

b. --list-rich-rules='<RULE>'

c. --list-rich-rules

d. --query-rich-rule

17. For use on computers in the demilitarized zone. Only selected incoming connections are accepted, and limited access to the internal network is allowed.

a. Dmz

b. Drop

c. External

d. Home

18. What from the following isn't in the Firewalld services configuration files?

a. Service ID

b. Port

c. Protocol

d. Short description

19. To remove a service from the firewalld configuration:

a. --erase-service=<service-name>

b. --delete-service=<service-name>

c. --remove-service=<service-name>

d. --disable-port=<service-name>

20. To create a service configuration file:
    a. Copy and modify a template file from /usr/lib/firewalld/services
    b. Use the firewall-cmd --add-service command
    c. Use the firewall-cmd --new-service command
    d. Rename and customize a template file in /usr/lib/firewalld/services

Answers to chapter Review Questions:

1. a
2. c
3. d
4. c
5. c
6. b
7. b
8. d
9. b
10. a
11. a
12. a
13. a
14. b
15. a
16. c
17. a
18. a
19. c
20. a, c