Workshop – How To Set Up a Firewall Using FirewallD

Part1: Introduction

Net-filter as we all know it's a firewall in Linux. Firewalld is a dynamic daemon to manage firewall with support for networks zones. In earlier version, RHEL & CentOS 6 we have been using iptables as a daemon for packet filtering framework. In RHEL/CentOS 7 and Fedora 21 iptables interface is being replaced by firewalld. In this guide, we will cover how to set up a firewall for your server and show you the basics of managing the firewall with the **firewall-cmd** administrative tool.

Basic Concepts in Firewalld:

Zones: The firewalld daemon manages groups of rules using entities called "zones". Zones are basically sets of rules dictating what traffic should be allowed depending on the level of trust you have in the networks your computer is connected to. Network interfaces are assigned a zone to dictate the behavior that the firewall should allow.

For computers that might move between networks frequently (like laptops), this kind of flexibility provides a good method of changing your rules depending on your environment. You may have strict rules in place prohibiting most traffic when operating on a public WiFi network, while allowing more relaxed restrictions when connected to your home network. For a server, these zones are not as immediately important because the network environment rarely, if ever, changes.

Regardless of how dymaic your network environment may be, it is still useful to be familiar with the general idea behind each of the pre-defined zones for firewalld. In order from least trusted to most trusted, the pre-defined zones within firewalld are:

drop: The lowest level of trust. All incoming connections are dropped without reply and only outgoing connections are possible.

block: Similar to the above, but instead of simply dropping connections, incoming requests are rejected with an icmp-host-prohibited or icmp6-adm-prohibited message.

public: Represents public, untrusted networks. You don't trust other computers but may allow selected incoming connections on a case-by-case basis.

external: External networks in the event that you are using the firewall as your gateway. It is configured for NAT masquerading so that your internal network remains private but reachable.

internal: The other side of the external zone, used for the internal portion of a gateway. The computers are fairly trustworthy and some additional services are available.

dmz: Used for computers located in a DMZ (isolated computers that will not have access to the rest of your network). Only certain incoming connections are allowed.

work: Used for work machines. Trust most of the computers in the network. A few more services might be allowed.

home: A home environment. It generally implies that you trust most of the other computers and that a few more services will be accepted.

trusted: Trust all of the machines in the network. The most open of the available options and should be used sparingly.

Permanence:

In firewalld, rules can be designated as either permanent or immediate. If a rule 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.

Part2: Turning on the Firewall

Before we can begin to create our firewall rules, we need to actually turn the daemon on. The **systemdunit** file is called **firewalld.service**. We can start the daemon for this session by typing:

 # systemctl start firewalld.service

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

 # firewall-cmd --state

Part 3: Getting Familiar with the Current Firewall Rules

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

 # firewall-cmd --get-default-zone

 # firewall-cmd --get-active-zones

 # firewall-cmd --list-all

Part 4: Exploring Alternative 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

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

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

Part 5: Selecting Zones for your Interfaces

Unless you have configured your network interfaces otherwise, each interface will be put in the default zone when the firewall is booted.

You can transition an interface between zones during a session by using the --zone= parameter in combination with the --change-interface= parameter.

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

 # firewall-cmd --zone=home --change-interface=eth0

Important Note:

Whenever you are transitioning an interface to a new zone, be aware that you are probably modifying the services that will be operational. For instance, here we are moving to the "home" zone, which has SSH available. This means that our connection shouldn't drop. Some other zones do not have SSH enabled by default and if your connection is dropped while using one of these zones, you could find yourself unable to log back in.

We can verify that this was successful by asking for the active zones again:

 # firewall-cmd --get-active-zones

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

 # systemctl restart firewalld.service

 # firewall-cmd --get-active-zones

Part 6: Changing the Zone of your Interface Permanently

Interfaces will always revert to the default zone if they do not have an alternative zone defined within their configuration. On CentOS, these configurations are defined within the /etc/sysconfig/network-scriptsdirectory with files of the format ifcfg-**interface**.

 # echo ZONE=home >> /etc/sysconfig/network-scripts/ifcfg-eth0

To implement your changes, you'll have to restart the network service, followed by the firewall service:

 # systemctl restart network.service

 # sudo systemctl restart firewalld.service

After your firewall restarts, you can see that your eth0 interface is automatically placed in the "home" zone:

 # firewall-cmd --get-active-zones

Adjusting 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:

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

Part 7: Setting Rules for your Applications

Adding a Service to your Zones:

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

Note: You can get more details about each of these services by looking at their 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 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. By default, this will only adjust the current firewall session. You can adjust the permanent firewall configuration by including the **--permanent** flag.

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:

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

ou 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

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:

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

 # firewall-cmd --zone=public --permanent --list-services

 # firewall-cmd --zone=public --add-service=https

 # firewall-cmd --zone=public --permanent --add-service=https

Part 8: Opening a Port for your Zones

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:

 # firewall-cmd --zone=public --add-port=5000/tcp

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

 # firewall-cmd --list-ports

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:

 # 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:

 # firewall-cmd --zone=public --permanent --add-port=5000/tcp

 # firewall-cmd --zone=public --permanent --add-port=4990-4999/udp

 # firewall-cmd --zone=public --permanent --list-ports

Part 9: Defining a Service

Opening ports for your zones is easy, but it can be difficult to keep track of what each one is for. If you ever decommission a service on your server, you may have a hard time remembering which ports that have been opened are still required. To avoid this situation, it is possible to define a service.

Services are simply collections of ports with an associated name and description. Using services is easier to administer than ports, but requires a bit of upfront work. The easiest way to start is to copy an existing script (found

in **/usr/lib/firewalld/services**) to the **/etc/firewalld/services** directory where the firewall looks for non-standard definitions.

For instance, we could copy the SSH service definition to use for our "example" service definition like this. The filename minus the **.xml** suffix will dictate the name of the service within the firewall services list:

cp /usr/lib/firewalld/services/service.xml /etc/firewalld/services/anisa.xml

Change the parameter as below:

```
<?xml version="1.0" encoding="utf-8"?>
<service>
  <short>Anisa</short>
  <description>This is a test.</description>
  <port protocol="tcp" port="7777"/>
  <port protocol="udp" port="8888"/>
</service>
```

Reload your firewall to get access to your new service:

firewall-cmd --reload

You can see that it is now among the list of available services:

firewall-cmd --get-services

Part 10: Defining a Service

While the predefined zones will probably be more than enough for most users, it can be helpful to define your own zones that are more descriptive of their function.

For instance, you might want to create a zone for your web server, called "publicweb". However, you might want to have another zone configured for the DNS service you provide on your private network. You might want a zone called "privateDNS" for that.

When adding a zone, you must add it to the permanent firewall configuration. You can then reload to bring the configuration into your running session. For instance, we could create the two zones we discussed above by typing:

firewall-cmd --permanent --new-zone=publicweb

 # firewall-cmd --permanent --new-zone=privateDNS

 # firewall-cmd --permanent --get-zones

 # firewall-cmd --get-zones

Reload the firewall to bring these new zones into the active configuration:

 # firewall-cmd --reload

 # firewall-cmd --get-zones

For instance, for the "publicweb" zone, you might want to add the SSH, HTTP, and HTTPS services:

 # firewall-cmd --zone=publicweb --add-service=ssh

 # firewall-cmd --zone=publicweb --add-service=http

 # firewall-cmd --zone=publicweb --add-service=https

 # firewall-cmd --zone=publicweb --list-all

Likewise, we can add the DNS service to our "privateDNS" zone:

 # firewall-cmd --zone=privateDNS --add-service=dns

 # firewall-cmd --zone=privateDNS --list-all

At this point, you have the opportunity to test your configuration. If these values work for you, you will want to add the same rules to the permanent configuration. You can do that by re-applying the rules with the --permanent flag:

 # firewall-cmd --zone=publicweb --permanent --add-service=ssh

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

 # firewall-cmd --zone=publicweb --permanent --add-service=https

 # firewall-cmd --zone=privateDNS --permanent --add-service=dns

Part 11: Enable Your Firewall to Start at Boot

At the beginning of the guide, we started our firewalld service, but we did not enable it. If you are happy with your current configuration and have tested that it is functional when you restart the service, you can safely enable the service.

To configure your firewall to start at boot, type:

 # systemctl enable firewalld

When the server restarts, your firewall should be brought up, your network interfaces should be put into the zones you configured (or fall back to the configured default zone), and the rules associated with the zone(s) will be applied to the associated interfaces.

Best Regards ☺

Fanavaran Anisa - 2017