7.Manage security

1. **Configure firewall settings using firewall-config, firewall-cmd, or iptables**

**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](https://www.certdepot.net/rhel7-use-sysctl/).

Although **Firewalld** is the **RHEL 7** way to deal with firewalls and provides many improvements, [iptables can still be used](https://www.certdepot.net/rhel7-disable-firewalld-use-iptables/) (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](https://lwn.net/Articles/484506/) 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](http://www.advancedclustering.com/act_kb/fixing-firewall-zones-centos-7-5/)).  
Note2: More information about the **nmcli** command is available at the [page dedicated to nmcli](https://www.certdepot.net/rhel7-get-started-nmcli/) or at the [IPV4 configuration page](https://www.certdepot.net/rhel7-configure-ipv4-addresses/).  
Note3: The **RHEL 7.3** release improves the way **Firewalld** handles zones (v0.3.9 -> v0.4.3.2: BZ#[1302802](https://bugzilla.redhat.com/show_bug.cgi?id=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](https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/Security_Guide/sec-Using_Firewalls.html)). This means that if you plan to stop **NetworkManager** for any reason (for example when [building a **KVM** host](https://www.certdepot.net/rhel7-configure-lab-network-settings/)), you will have to [stop Firewalld and use Iptables instead](https://www.certdepot.net/rhel7-disable-firewalld-use-iptables/)!  
Note: With the **RHEL 7.3** release, **Firewalld** robustness has been improved in regard to **NetworkManager** (see details [here](http://www.firewalld.org/2016/05/firewalld-0-4-2-release)).

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](http://serverfault.com/questions/680780/block-all-but-a-few-ips-with-firewalld" \o "http://serverfault.com/questions/680780/block-all-but-a-few-ips-with-firewalld).

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](https://bertvv.github.io/presentation-el7-basics/)):

# **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](http://ktaraghi.blogspot.fr/2013/10/what-is-firewalld-and-how-it-works.html" \o "http://ktaraghi.blogspot.fr/2013/10/what-is-firewalld-and-how-it-works.html).  
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](https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/Security_Guide/sec-Using_Firewalls.html#sec-Configuring_firewalld)). 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](https://www.certdepot.net/rhel7-rc-local-service/), 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](http://unix.stackexchange.com/questions/240044/on-centos7-firewalld-overwrite-iptables-modules" \o "http://unix.stackexchange.com/questions/240044/on-centos7-firewalld-overwrite-iptables-modules).

## **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,](http://searchdatacenter.techtarget.com/tip/A-few-ways-to-configure-Linux-firewalld)
* watch **Thomas Woerner**‘s video about [Firewalld, present and future (48min/2015)](https://www.youtube.com/watch?v=L8rwSqONmCY" \l "t=4m45s" \o "https://www.youtube.com/watch?v=L8rwSqONmCY#t=4m45s),
* read this [FedoraProject page about Fail2ban with Firewalld](https://fedoraproject.org/wiki/Fail2ban_with_FirewallD),
* read this article about [Firewalld and zone deployment by James Hogarth](https://www.hogarthuk.com/?q=node/9),
* read the [CIS RHEL 7 Server Hardening Guide](https://benchmarks.cisecurity.org/tools2/linux/CIS_Red_Hat_Enterprise_Linux_7_Benchmark_v1.1.0.pdf),
* watch **Venkat Nagappan**‘s video about [Firewalld Concepts and Examples (34min/2015)](https://www.youtube.com/watch?v=TyMallqnWiw),
* watch **Sander van Vugt**‘s video about [port forwarding using firewall-cmd (8min/2015)](https://www.youtube.com/watch?v=MKTDFkwgMRQ),
* watch **Ralph Nyberg**‘s video about [Firewalld and Iptables (26min/2016)](https://www.youtube.com/watch?v=XF9sjjLM8_0),
* read **Thomas Woerner**‘s blog, the [blog of the author of Firewalld](http://www.firewalld.org/),
* read this presentation from the [11th Netfilter Workshop (2015)](http://www.firewalld.org/uploads/2015/06/nfws2015-firewalld.pdf),
* read the [changelog of the Firewalld versions](https://github.com/t-woerner/firewalld/releases),
* have a look at **Daniel Aleksandersen**‘s blog about [Configuring zones bound by source IPs in Firewalld](https://www.ctrl.blog/entry/how-to-firewalld-zone-by-ip) or [Comparing and contrasting Uncomplicated Firewall and Firewalld](https://www.ctrl.blog/entry/ufw-vs-firewalld),
* read **Alexander Molochko**‘s blog about [Creating a new zone in Firewalld](https://crosp.net/blog/administration/drop-all-accept-few-zone-firewalld-centos-7/).

Sources: [RHEL7 Security Guide](https://access.redhat.com/site/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/Security_Guide/sec-Using_Firewalls.html), [wiki Fedora project](https://fedoraproject.org/wiki/FirewallD).

1. **Configure key-based authentication for SSH**

Instead of connecting through login/password to a remote host, **SSH** allows you to use key-based authentication. To set up key-based authentication, you need two virtual/physical servers that we will call **server1** and **server2**.

## **Configuration Procedure**

On the **server1**, create a user **user01** with password **user01**:

# **useradd user01**

# **passwd user01**

Changing password for user user01.

New password:

Retype new password:

passwd: all authentication tokens updated successfully.

On the **server2**, create the same user with password **user01**:

# **useradd user01**

# **passwd user01**

Changing password for user user01.

New password:

Retype new password:

passwd: all authentication tokens updated successfully.

On the **server1**, connect as this new user:

# **su - user01**

Generate a private/public pair for key-based authentication (here **rsa** key with **2048** bits and **no passphrase**):

[user01@server1 ~]$ **ssh-keygen -b 2048 -t rsa**

Generating public/private rsa key pair.

Enter file in which to save the key (/home/user01/.ssh/id\_rsa): **return**

Created directory '/home/user01/.ssh'.

Enter passphrase (empty for no passphrase): **return**

Enter same passphrase again: **return**

Your identification has been saved in /home/user01/.ssh/id\_rsa.

Your public key has been saved in /home/user01/.ssh/id\_rsa.pub.

The key fingerprint is:

6d:ac:45:32:34:ac:da:4a:3b:4e:f2:83:85:84:5f:d8 user01@server1.example.com

The key's randomart image is:

+--[ RSA 2048]----+

| .o |

| ... |

| . o .o . |

|. o E . \* |

| o o o S = |

| o + . + |

| .+.o . |

| .+= |

| .oo |

+-----------------+

Still on **server1**, copy the public key to **server2**.

[user01@server1 ~]$ **ssh-copy-id -i .ssh/id\_rsa.pub user01@server2.example.com**

The authenticity of host 'server2.example.com (192.168.1.49)' can't be established.

ECDSA key fingerprint is 67:79:67:88:7f:da:31:49:7b:dd:ed:40:af:ae:b6:ae.

Are you sure you want to continue connecting (yes/no)? yes

/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed

/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys

user01@server2.example.com's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'user01@server2.example.com'"

and check to make sure that only the key(s) you wanted were added.

On the **server2**, edit the **/etc/ssh/sshd\_config** file and set the following options:

PasswordAuthentication no

PubkeyAuthentication yes

Note: Don’t hesitate to [set up a virtual console access](https://www.certdepot.net/rhel7-access-virtual-machines-console/) on **server2**, this will avoid re-installing the physical/virtual server if something goes wrong.

Restart the **sshd** service:

# **systemctl restart sshd**

## **Testing Time**

On the **server1** as **user01**, connect to the **server2**:

[user01@server1 ~]$ **ssh server2.example.com**

Note1: This configuration can also be done for the **root** account.  
Note2: Use **-v**, **-vv**, or **-vvv** options to get some debug information.

## **Additional Resources**

**Bob Cromwell** wrote a series of articles about [setting up SSH keys for easier and more secure authentication](http://blog.learningtree.com/set-ssh-keys-easier-secure-authentication/), [setting up a SSH key-agent](http://blog.learningtree.com/how-set-up-ssh-key-agent/), [easily maintaining multiples websites with SSH](http://blog.learningtree.com/easily-maintain-multiple-websites-ssh/) and [ways to manage your SSH keys and identities](http://blog.learningtree.com/ways-manage-ssh-keys-identities/).

Beyond the exam objectives, **Scott Lowe** explains how to [build a bastion SSH](http://blog.scottlowe.org/2017/05/26/bastion-hosts-custom-ssh-configs/).

1. **Set enforcing and permissive modes for SELinux**

**SELinux** stands for **S**ecurity-**E**nhanced **Linux**. It is a way to improve the server security.

The **/etc/selinux/config** file stores the current configuration:

# more /etc/selinux/config

# This file controls the state of SELinux on the system.

# SELINUX= can take one of these three values:

#     enforcing - SELinux security policy is enforced.

#     permissive - SELinux prints warnings instead of enforcing.

#     disabled - No SELinux policy is loaded.

SELINUX=enforcing

# SELINUXTYPE= can take one of three two values:

#     targeted - Targeted processes are protected,

#     minimum - Modification of targeted policy. Only selected processes are protected.

#     mls - Multi Level Security protection.

SELINUXTYPE=targeted

**SELinux** can run in three different modes (**enforcing**, **permissive** and **disabled**) well described in the above file.

Besides the mode, there is a **SELinux** type (**targeted**, **minimum** and **mls**). Except if you work in a military agency, you will never need to change the **targeted** type.

## **Configuration**

To get the current **SELinux** status:

# sestatus

To set **enforcing** mode, type:

# setenforce enforcing

To make this change permanent, edit the **/etc/sysconfig/selinux** file (or the **/etc/selinux/config**file) and replace the following value:

SELINUX=**enforcing**

**Alternatively**, to set **permissive** mode, type:

# setenforce permissive

To make this change permanent, edit the **/etc/sysconfig/selinux** file (or the **/etc/selinux/config**file) and replace the following value:

SELINUX=**permissive**

To make the reboot mandatory to change the configuration (**-P** can be added but **with caution**), type:

# setsebool **secure\_mode\_policyload** on

## **Additional Resources**

Also, you can:

* watch **Thomas Cameron’s 2015 Red Hat Summit** presentation [SELinux for mere mortals (52min/2015)](https://www.youtube.com/watch?v=cNoVgDqqJmM),
* follow [Red Hat Jamie Duncan’s SELinux workshop](http://people.redhat.com/~jduncan/workshops/selinux-vcu/),
* read [Sven Vermeulen’s blog](http://blog.siphos.be/) to better understand **SELinux**,
* buy [Sven Vermeulen’s book](http://www.amazon.com/SELinux-System-Administration-Sven-Vermeulen/dp/1783283173) to get a complete presentation about **SELinux**,
* follow the [Gentoo SELinux tutorials](https://wiki.gentoo.org/wiki/SELinux/Tutorials) written by **Sven Vermeulen**,
* consult the [SELinux Userspace wiki](https://github.com/SELinuxProject/selinux/wiki),
* read the [CIS RHEL 7 Server Hardening Guide](https://benchmarks.cisecurity.org/tools2/linux/CIS_Red_Hat_Enterprise_Linux_7_Benchmark_v1.1.0.pdf),
* read [Limestone Networks’](https://www.limestonenetworks.com/) page about [hardening CentOS](https://www.limestonenetworks.com/support/knowledge-center/11/83/hardening_centos.html),
* have a look at the [SELinux Game website](http://selinuxgame.org/),
* read **Thomas Cameron’s 2018 Red Hat Summit** presentation about [SEcurity ENhanced Linux for Mere Mortals](http://people.redhat.com/tcameron/Summit2018/selinux/SELinux_for_Mere_Mortals_Summit_2018.pdf),
* read [How SELinux helps mitigate risk while facilitating compliance](https://access.redhat.com/blogs/766093/posts/3557091).

1. **List and identify SELinux file and process context**

To get a **SELinux** file context, type:

# ls -Z

To get a **SELinux** process context, type:

# ps -eZ

1. **Restore default file contexts**

To restore the default **SELinux** file contexts at a specified location (here **/path**), type:

# restorecon -R /path

1. **Use boolean settings to modify system SELinux settings**

**SELinux** uses booleans to makes its policy more flexible.

The basic policy is pretty strict but suits most requirements. But if you’ve got special needs, you can easily adjust it thanks to the **SELinux** booleans.

## **Standard Management**

Get the list of **SELinux** booleans on a particular topic (here **ftp**):

# getsebool -a | grep **ftp**

allow\_ftpd\_anon\_write --> off

allow\_ftpd\_full\_access --> on

allow\_ftpd\_use\_cifs --> off

allow\_ftpd\_use\_nfs --> off

ftp\_home\_dir --> on

ftpd\_connect\_db --> off

ftpd\_use\_fusefs --> off

ftpd\_use\_passive\_mode --> off

httpd\_enable\_ftp\_server --> off

tftp\_anon\_write --> off

tftp\_use\_cifs --> off

tftp\_use\_nfs --> off

To set a specific **SELinux** boolean according to your need (here **ftp\_home\_dir**), type:

# setsebool -P **ftp\_home\_dir** on

or

# yum install -y **setroubleshoot-server**

# semanage boolean -m --on **ftp\_home\_dir**

Note1: You can use **on** or **1**, **off** or **0**with the **setsebool** command.  
Note2: The **-P** option means **P**ermanent. If you don’t use it, the boolean will restore its previous permanent or default configuration after the next reboot.  
Note3: The **semanage boolean** command only assigns permanent configurations.

To get a more detailed list of **SELinux** booleans, type:

# yum install -y **setroubleshoot-server**

# semanage boolean -l

SELinux boolean                State  Default Description

ftp\_home\_dir                   (off  ,  off)  Allow ftp to read and write files in the user home directories

smartmon\_3ware                 (off  ,  off)  Enable additional permissions needed to support devices on 3ware controllers.

xdm\_sysadm\_login               (off  ,  off)  Allow xdm logins as sysadm

xen\_use\_nfs                    (off  ,  off)  Allow xen to manage nfs files

mozilla\_read\_content           (off  ,  off)  Control mozilla content access

ssh\_chroot\_rw\_homedirs         (off  ,  off)  Allow ssh with chroot env to read and write files in the user home directories

postgresql\_can\_rsync           (off  ,  off)  Allow postgresql to use ssh and rsync for point-in-time recovery

allow\_console\_login            (on   ,   on)  Allow direct login to the console device. Required for System 390

spamassassin\_can\_network       (off  ,  off)  Allow user spamassassin clients to use the network.

authlogin\_shadow               (off  ,  off)  Allow users login programs to access /etc/shadow.

httpd\_can\_network\_relay        (off  ,  off)  Allow httpd to act as a relay

openvpn\_enable\_homedirs        (on   ,   on)  Allow openvpn to read home directories

...

**Important note**: The **State** column respectively shows the **current** boolean configuration and the **Default** column the **permanent** boolean configuration.

To get the list of all the **SELinux** booleans with a current value different from the **default**one (**-C**option for local **C**ustomization), type:

# semanage boolean -l -C

SELinux boolean State Default Description

ftp\_home\_dir (on , on) Allow ftp to read and write files in the user home directories

httpd\_can\_sendmail (on , on) Allow http daemon to send mail

allow\_postfix\_local\_write\_mail\_spool (on , on) Allow postfix\_local domain full write access to mail\_spool directories

allow\_ftpd\_full\_access (on , on) Allow ftp servers to login to local users and read/write all files on the system, governed by DAC.

To display the list of **SELinux** booleans related to **NFS**, type:

# semanage boolean -l | egrep "nfs|SELinux"

SELinux boolean State Default Description

xen\_use\_nfs (off , off) Allow xen to use nfs

virt\_use\_nfs (off , off) Allow virt to use nfs

mpd\_use\_nfs (off , off) Allow mpd to use nfs

nfsd\_anon\_write (off , off) Allow nfsd to anon write

ksmtuned\_use\_nfs (off , off) Allow ksmtuned to use nfs

git\_system\_use\_nfs (off , off) Allow git to system use nfs

virt\_sandbox\_use\_nfs (off , off) Allow virt to sandbox use nfs

logrotate\_use\_nfs (off , off) Allow logrotate to use nfs

git\_cgi\_use\_nfs (off , off) Allow git to cgi use nfs

cobbler\_use\_nfs (on , off) Allow cobbler to use nfs

httpd\_use\_nfs (off , off) Allow httpd to use nfs

sge\_use\_nfs (off , off) Allow sge to use nfs

ftpd\_use\_nfs (off , off) Allow ftpd to use nfs

sanlock\_use\_nfs (off , off) Allow sanlock to use nfs

samba\_share\_nfs (off , off) Allow samba to share nfs

openshift\_use\_nfs (off , off) Allow openshift to use nfs

polipo\_use\_nfs (off , off) Allow polipo to use nfs

use\_nfs\_home\_dirs (off , off) Allow use to nfs home dirs

nfs\_export\_all\_rw (on , on) Allow nfs to export all rw

nfs\_export\_all\_ro (on , on) Allow nfs to export all ro

## **Additional Resources**

You can also watch this video from [Sander van Vugt](http://www.rhatcertification.com/) about [Understanding SELinux Booleans (10min/2014)](https://www.youtube.com/watch?v=5WiI70dzmSE).

1. **Diagnose and address routine SELinux policy violations**

## **Introduction**

In **RHEL 7**, each package doesn’t store its own **SELinux** policy. The **SELinux** policy is stored in one and only one package called **selinux-policy-targeted**.

When a policy has been written for a given process and **SELinux** is in **Permissive** or **Enforcing**mode, all action not allowed by the **SELinux** policy will trigger a violation.

The following procedure will give you some details about any **SELinux** policy violation.

## **Main Procedure**

Install the **setroubleshoot-server** package:

# yum install -y setroubleshoot-server

Note: In fact, it’s the **policycoreutils-python** package that really contains the **semanage**command. However, I have always found the **setroubleshoot-server** package name, that contains the **policycoreutils-python** package itself, easier to remember!

Display the **SELinux** policy violations:

# sealert -a /var/log/audit/audit.log

In addition, when an **AVC** (**A**ccess **V**ector **C**ache) event occurs, you can grab the associated line displayed in the **/var/log/audit/audit.log** file and send it to the **audit2why** command to get a diagnostic.

For example, let’s assume you’ve got this line in your **/var/log/audit/audit.log** file:

type=AVC msg=audit(**1415714880.156:29**): avc: denied { name\_connect } for pid=1349 \

comm="nginx" dest=8080 scontext=unconfined\_u:system\_r:httpd\_t:s0 \

tcontext=system\_u:object\_r:http\_cache\_port\_t:s0 tclass=tcp\_socket

Execute this command to get a diagnostic:

# **grep 1415714880.156:29 /var/log/audit/audit.log | audit2why**

Was caused by:

One of the following booleans was set incorrectly.

Description:

Allow httpd to act as a relay

Allow access by executing:

# setsebool -P httpd\_can\_network\_relay 1

Description:

Allow HTTPD scripts and modules to connect to the network using TCP.

Allow access by executing:

# setsebool -P httpd\_can\_network\_connect 1

This will make your investigation much easier!

## **Additional Resources**

[Jens Depuydt’s blog](http://jensd.be/) provides a good article called [SELinux in a practical way](http://jensd.be/?p=640) about this topic.  
[Sander van Vugt](http://www.rhatcertification.com/) offers an interesting video about [Fixing SELinux Issues (48min/2015)](https://www.youtube.com/watch?v=TeaupdLYzmE).  
In addition, **Red Hat** provides a video about [Monitoring SELinux Violations (10min/2016)](https://www.youtube.com/watch?v=obN27RcHaao).  
During the **2016 DevConf.cz** a presentation was given about the [Big SElinux Troubleshooting Chart (95min/2016)](https://www.youtube.com/watch?v=SR105KR2Tc0) (pdf [here](http://bit.ly/1oxz1fO)).

Beyond the exam objectives, you could be interested in this post from **Dan Walsh** about [SELinux Users and Roles](http://danwalsh.livejournal.com/75683.html).  
The [mgrepl website](https://mgrepl.wordpress.com/) also provides very interesting articles about **SELinux** security policy