Red Hat Enterprise RHCSA | RHCSE Preparation Documentation

Release 0.0.1

Herwig Gans

Apr 02, 2018

Contents

CHAPTER 1

Introduction

1. I prefer documentation in text files
2. I like to store files offline and online
3. I need constant backup of my documentation

Reason enough to use Sphinx together with github and Red The Docs !

Structured Text Help:: http://www.sphinx-doc.org/en/stable/rest.html#rst-primer http://openalea.gforge.inria.fr/doc/openalea/doc/_build/html/source/sphinx/rest_syntax.html

RHEL Product Documentation:: https://access.redhat.com/documentation/en/red-hat-enterprise-linux/

Offline documentation besides man pages:: /usr/share/doc/

1.1 Installation

1.1.1 CentOs Installation

Force Text Setup

You can also force installation in text mode if you wish. To do so, highlight the Install Red Hat Enterprise Linux 7.0 option and press the tab key. When you do, the following options are revealed on that screen, on one line:

```
> vmlinuz initrd=initrd.img inst.stage2=hd:LABEL=RHEL-7.0\x20Server.x86_64 quiet
```

To force installation in text mode, add **inst.text** to the end of this line.

How to setup a local yum repository for locally mounted DVD/ISO

Documented under commands-collection yum: *How to setup a local yum repository for locally mounted ISO*.

How to install Display Manager (GNOME) after initial (minimal) Installation

```
# yum group install "Server with GUI"
# systemctl get-default
  multi-user.target
# systemctl set-default graphical.target
  Removed symlink /etc/systemd/system/default.target.
  Created symlink from /etc/systemd/system/default.target to /usr/lib/systemd/system/
↪graphical.target.
```

The command systemctl **set-default graphical.target** is important that the GUI starts with system boot. It brings the system from runlevel 3 to runlevel 5

Indices and tables

- genindex
- modindex
- search

1.1.2 kickstart

Two methods for creating the kickstart configuration file:

- /root/anaconda-ks.cfg
- graphical kickstart configurator **system-config-kickstart** command

```
# yum install system-config-kickstart
```

There is also software acting as installation server for greater deployments like: http://cobbler.github.io/

How to start installation with kickstart file

1. Mount filesystem with kickstart file
2. On the first menu of RHEL installation press <TAB>
3. Add the "ks" option to the vmlinz parameter line

```
> vmlinuz initrd=initrd.img inst.stage2=hd:LABEL=RHEL-7.0 \x20Server.x86_64 quiet␣
↪ks=hd:sdb1:/ks.cfg
```

It is possible to fetch the kickstart file from network too

```
ks=nfs:192.168.122.1:/ks.cfg
ks=http://192.168.122.1/ks.cfg
```

Important kickstart file options

After kickstart configuration file creation a validation should be made via **ksvalidator** command.

```
# version=RHEL7
# authconfig --enableshadow --passalgo=sha512
```

Installation source:

```
 # cdrom
 # nfs --server=192.168.122.1 --dir=/inst
 # url --url http://192.168.122.1/inst
 # url --url ftp://192.168.122.1/pub/inst

 # --> you can also point the install source to an ISO file:
 # harddrive --partition=/dev/sda10 --dir=/tmp/michael/

 # firstboot --disabled

 # keyboard --vckeymap=us --xlayouts='us'
 # lang en_US.UTF-8

 # --> you can use the password from /etc/shadow because it uses same encryption
 # rootpw --iscrypted $6$5UrLfXTk$CsCW0nQytrUuvycuLT317/
 # user --groups=wheel name=michael --password=... --iscrypted --gecos="MJ"

# timezone America/Los_Angeles --isUtc
```

The network configuration must be **one line**!

```
# network --device eth0 --bootproto dhcp
# network --bootproto static --device=eth0 --gateway=192.168.122.1 --ip=192.168.122.
↪150 --netmask=255.255.255.0 --noipv6 --nameserver==192.168.122.1 --activate network␣
↪--hostname tester1.example.com

# firewall --service=ssh
# selinux --enforcing
```

Boot storage and partitions Remove any –onpart directives, otherwise an error occurs during installation

```
# bootloader --location=mbr --boot-drive=vda
# clearpart --all --initlabel --drives=vda

# part /boot --fstype="xfs" --size=500
# part swap --fstype="swap" --size=1000
# part / --fstype="xfs" --size=10000
# part /home --fstype="xfs" --size=1000

# part pv.01 --fstype="lvmpv" --ondisk=vda --size=11008
# part /boot --fstype="xfs" --ondisk=vda --size=500
# part swap --fstype="swap" --ondisk=vda --size=1000
# volgroup rhel --pesize=4096 pv.01
# logvol / --fstype="xfs" --size=10000 --name=root --vgname=rhel
# logvol /home --fstype="xfs" --size=1000 -name=home --vgname=rhel

#repo --name="Red Hat Enterprise Linux" --baseurl=ftp://192.168.122.1/pub/inst --
↪cost=100
```

Package installation: What follows is a list of package groups that are installed through this Kickstart configuration file. These names correspond to the names you can find in the *-comps-Server.x86_64.xml file in the RHEL 7 DVD /repodata directory described in Chapter 1. Because the list is long, the following is just an excerpt of package groups (which startwith @) and package names

```
# %packages
# @base
# @core
# ...
# @print-client
# @x11
# %end

# %post
# ...
```

Working example for KVM based kickstart installation

It is important to have an ISO file as CentOS/RHEL source (cdrom passthrough to KVM VM is currently disabled by RH!), this is necessary to see the installation screen, it allows to alter the boot parameters (by pressing TAB) Second thing is to use “cdrom” as installation media for KVM installation, because network based installation is more complicated.

```
[root@kvm-server01 ~]# cat /var/www/html/inst/ks.cfg
#version=DEVEL
# License agreement
eula --agreed
# System authorization information
auth --enableshadow --passalgo=sha512

# Use CDROM installation media
cdrom
# url --url="http://192.168.122.1/inst"

# Use graphical install
graphical

# Run the Setup Agent on first boot
# firstboot --enable
firstboot --disabled

# System services
services --disabled="chronyd"

# Keyboard layouts
keyboard --vckeymap=at --xlayouts='at'

# System language
lang en_US.UTF-8

# ignoredisk --only-use=sda

# Network information
network  --bootproto=static --device=eth0 --gateway=192.168.100.1 --ip=192.168.100.
↪100 --nameserver=8.8.8.8,192.168.0.237    --netmask=255.255.255.0 --ipv6=auto --
↪activate
network  --hostname=outsider01.example.com

# Root password
rootpw --iscrypted $6$GfLyBLABLABLA
```

```
# System timezone
timezone Europe/Vienna --isUtc --nontp

user --name=user1 --password=$6$FlVBBLABLABLA    --iscrypted --gecos="user1"

firewall --service=ssh
selinux --enforcing

# System bootloader configuration
bootloader --append=" crashkernel=auto" --location=mbr --boot-drive=vda

# Partition clearing information
clearpart --all --initlabel --drives=vda

# Disk partitioning information
part pv.01 --fstype="lvmpv" --ondisk=vda --size=13523
part /boot --fstype="xfs" --ondisk=vda --size=953
part swap --fstype="swap" --ondisk=vda --size=1907
volgroup centos --pesize=4096 pv.01
logvol /  --fstype="xfs" --size=9536 --name=root --vgname=centos
logvol /home  --fstype="xfs" --size=3979 --name=home --vgname=centos

# repo --name=myrepo --baseurl=http://192.168.122.1/inst

shutdown

%packages
@base
@core
%end

%addon com_redhat_kdump --enable --reserve-mb=auto
%end

%anaconda
pwpolicy root --minlen=6 --minquality=1 --notstrict --nochanges --notempty
pwpolicy user --minlen=6 --minquality=1 --notstrict --nochanges --emptyok
pwpolicy luks --minlen=6 --minquality=1 --notstrict --nochanges --notempty
%end
```

1.1.3 Indices and tables

- genindex
- modindex
- search

1.1.4 kvm

Does my CPU support Virtualization and is it enabled in BIOS / UEFI?

```
[gans@server1 ~]$ grep -c vmx /proc/cpuinfo
1
[gans@server1 ~]$
```

```
[gans@server1 ~]$ lsmod | grep kvm
kvm_intel              170086  0
kvm                    566340  1 kvm_intel
irqbypass               13503  1 kvm
```

How to run nested KVM within VMWare ESXi (5.5)

https://forum.ivorde.com/kvm-nested-in-vmware-esxi-5-5-enable-guest-hypervisor-vmx-svm-flags-without-vsphere-web-client-t1977
html

1. Shutdown the VM

2. Locate the guest hypervisor virtual machine configuration file (<VM-name.vmx>), edit and add the following line at the end

```
vhv.enable = "TRUE"
```

3. Identify the nested hypervisor vm ID and reload it's configuration with the **vim-cmd esxi command**

```
~ # vim-cmd vmsvc/getallvms | grep -i 50_s1
   30      RH_Lab122_50_s1           [datastore3] RH_Lab122_50_s1/RH_Lab122_50_s1.vmx ␣
↪          rhel6_64Guest             vmx-10
~ #
~ # vim-cmd vmsvc/reload 30
```

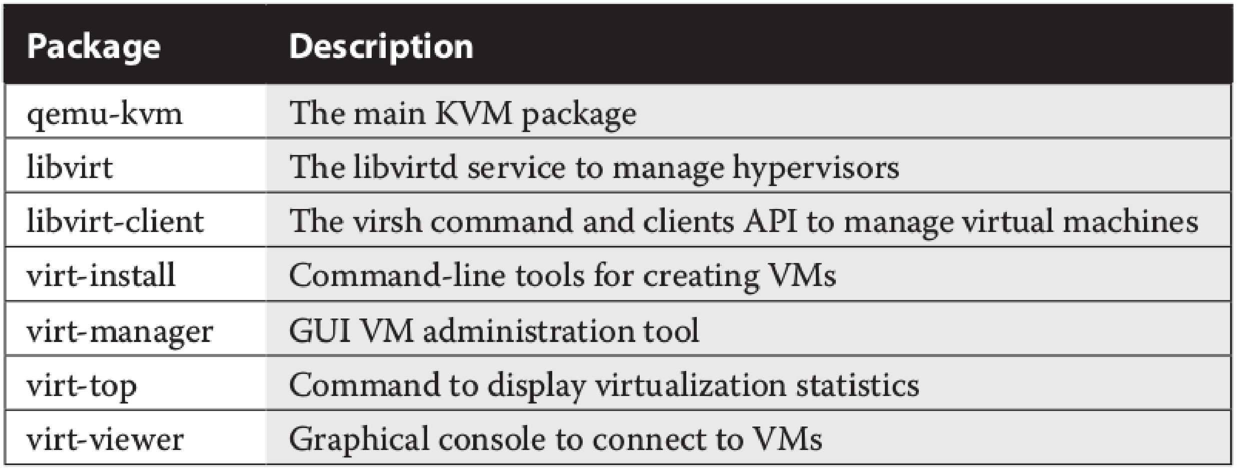
4. Start the VM

5. Verify that the nested hypervisor correctly detects the vmx flag

```
[gans@server1 ~]$ grep -c vmx /proc/cpuinfo
1
```

Packages to install for KVM in RHEL

These can be installed via the GNOME Software Manager



Package	Description
qemu-kvm	The main KVM package
libvirt	The libvirtd service to manage hypervisors
libvirt-client	The virsh command and clients API to manage virtual machines
virt-install	Command-line tools for creating VMs
virt-manager	GUI VM administration tool
virt-top	Command to display virtualization statistics
virt-viewer	Graphical console to connect to VMs

As an alternative, install the Virtualization Host and Virtualization Client groups:

```
# yum group install "Virtualization Host" "Virtualization Client"
```

Move KVM Storage location to another place

One advantage of this setup is that it retains the default SELinux settings for the /var /lib/libvirt/images directory, as defined in the file_contexts file in the /etc/selinux/targeted /contexts/files directory. In other words, this configuration survives a relabel of SELinux

```
# mkdir /home/gans/KVM
# su -
# semanage fcontext -a -t virt_image_t '/home/gans/KVM(/.*)?'
# restorecon /home/gans/KVM
# rmdir /var/lib/libvirt/images
# ln -s /home/gans/KVM /var/lib/libvirt/images
```

File permission issues when changing the VM Storage location to another place

When installing the first VM there may be permission errors for the linked **/var/lib/libvirt/images** directory. In my case I linked to /home/gans/KVM - so best approach for me to overcome these errors was to run virt-manager as my own user.

In file **/etc/libvirt/qemu.conf** find the user and group parameters and uncomment | enter own user and group information.

```
vi /etc/libvirt/qemu.conf
systemctl restart libvirtd.service
```

Application documentation - admin, usertasks a.s.o

Documented under applications kvm: *Mounting installation media for VM installation*.

Indices and tables

- genindex
- modindex
- search

1.2 Networking

1.2.1 network-configuration-troubleshooting

Red Hat stores network information in the directory **/etc/sysconfig/network-scripts** and in **/etc/sysconfig/network**. Example file of **/etc/sysconfig/network-scripts/ifcfg-ens192**

```
TYPE="Ethernet"
PROXY_METHOD="none"
BROWSER_ONLY="no"
BOOTPROTO="none"
```

```
DEFROUTE="yes"
IPV4_FAILURE_FATAL="no"
IPV6INIT="yes"
IPV6_AUTOCONF="yes"
IPV6_DEFROUTE="yes"
IPV6_FAILURE_FATAL="no"
IPV6_ADDR_GEN_MODE="stable-privacy"
NAME="ens192"
UUID="25946d9c-b79c-415d-97e0-638bc017c62d"
DEVICE="ens192"
ONBOOT="yes"
IPADDR="192.168.122.100"
PREFIX="24"
GATEWAY="192.168.122.254"
DNS1="192.168.0.237"
DNS2="8.8.8.8"
IPV6_PRIVACY="no"
ZONE=public
```

After changes in ifcfg files, they are applied via Network Manager command

```
# nmcli con reload
# nmcli con down eth0
# nmcli con up eth0
```

Check the status of network service

```
# systemctl status network
```

Restart networking

```
# systemctl restart network
```

Network Manager status check

```
# systemctl status NetworkManager
```

Network Manager CLI

```
# nmcli dev status
```

Configuring VLAN Trunk With Sub-Interfaces

First thing is to make sure that the 801.1q kernel module is loaded:

```
# lsmod | grep 8021q
```

If not, load it and make it persistent:

```
# modprobe 8021q
# echo "8021q" > /etc/modules-load.d/8021q.conf
```

The physical interface is configured in **/etc/sysconfig/network-scripts/ifcfg-***

```
DEVICE=eth0
TYPE=Ethernet
```

```
BOOTPROTO=none
ONBOOT=yes
```

Configure the VLAN interface script in /etc/sysconfig/network-scripts. The configuration filename must be the physical interface plus a "." character plus the VLAN id number. For example, if the VLAN id is 10, and the physical network interface is eth0, then the configuration filename should be ifcfg-eth0.10, as the example below:

```
DEVICE=eth0.10
BOOTPROTO=none
ONBOOT=yes
IPADDR=14.1.1.31
NETMASK=255.255.255.0
USERCTL=no
NETWORK=14.1.1.0
VLAN=yes
```

```
# service network restart
```

Indices and tables

- genindex
- modindex
- search

1.2.2 network-commands

ifconfig | arp | netstat | route are "obsolete"

ifconfig

```
# ip link show
# ip addr show
# ip -s link

# ip addr add 192.168.122.150/24 dev eth0
# ip link set dev eth0 up|down
# ip link set dev device promisc on|off
# ip link set dev device mtu N
```

arp and route

```
# ip neigh show
# ip route
```

netstat

```
# ss -tuna4
```

Configure | Manipulate Network Adapter

```
ip addr add 192.168.122.150/24 dev eth0
ip addr flush dev eth0
ip link set dev eth0 up | down
ip link set dev eth0 promisc on | off
```

Network Manager

- **nmcli** command line tool
- **nmtui** text based graphical tool
- **nm-connection-editor** GTK3 based tool

Check if it is running

```
systemctl status NetworkManager
```

```
# nmcli con show
NAME    UUID                                  TYPE            DEVICE
ens192  25946d9c-b79c-415d-97e0-638bc017c62d  802-3-ethernet  ens192
virbr0  c9f5d0ca-a518-436c-adea-6db1c243c71f  bridge          virbr0
virbr1  8d8c5bff-2558-4ceb-b99a-e1aa14a120f5  bridge          virbr1
vnet0   605fe99d-1d3b-4242-8127-d172c7450def  tun             vnet0

# nmcli con add con-name "eth0-work" type ethernet ifname eth0
# nmcli con mod "eth0-work" ipv4.addresses "192.168.20.100/24 192.168.20.1"
# nmcli con mod "eth0-work" +ipv4.dns 192.168.20.1
```

Notify NetworkManager about changes in config files

```
nmcli con reload
nmcli con down eth0
nmcli con up eth0
```

Switch to the new connection profile

```
nmcli con up "eth0-work"
nmcli dev status
```

Prevent Autostart

```
# nmcli con mod "eth0-work" connection.autoconnect no
```

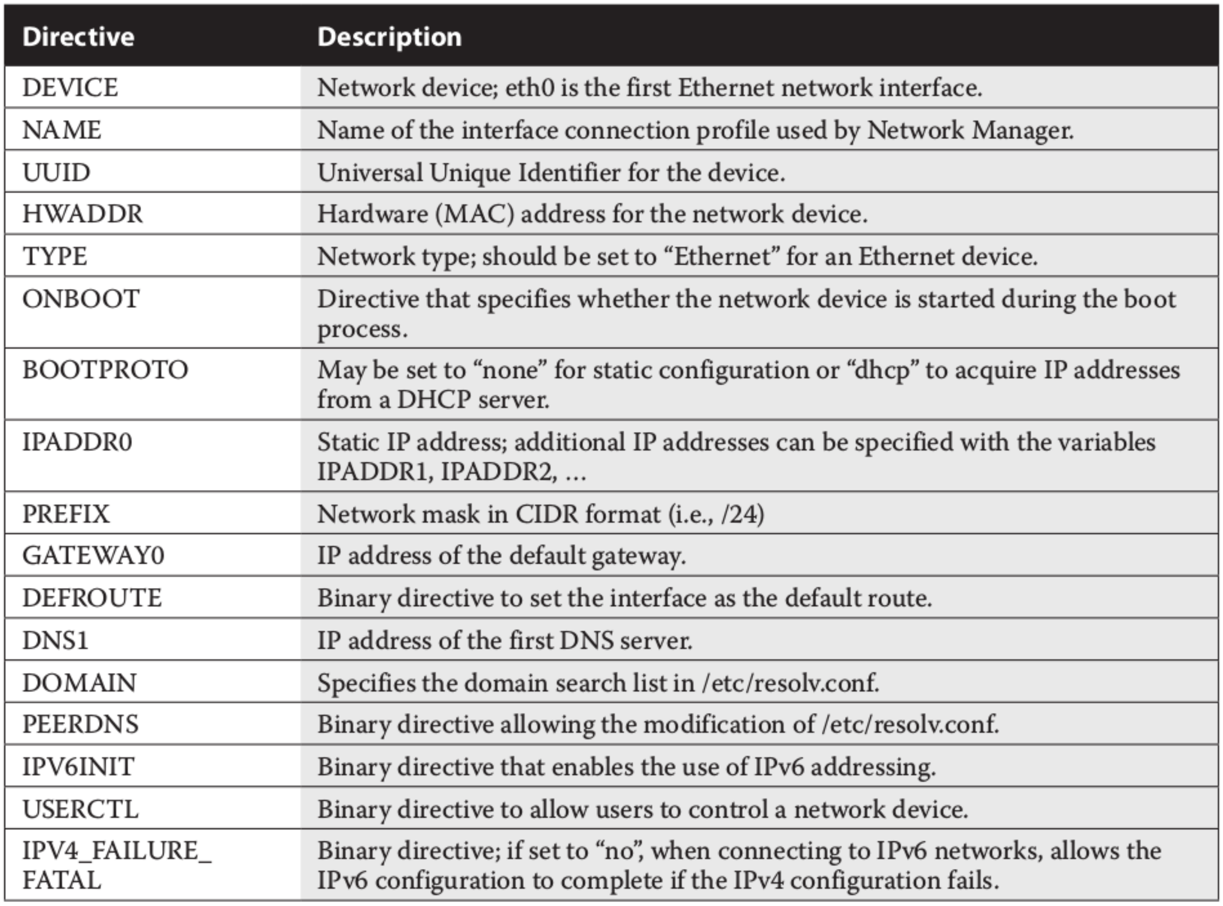
Graphical Networking User Interfaces

- nmtui
- nm-connection-editor

Network Connection Directives

These directives can be configured in interface scripts

TABLE 3-7 Network Configuration Directives in the /etc/sysconfig/network-scripts Directory



Directive	Description
DEVICE	Network device; eth0 is the first Ethernet network interface.
NAME	Name of the interface connection profile used by Network Manager.
UUID	Universal Unique Identifier for the device.
HWADDR	Hardware (MAC) address for the network device.
TYPE	Network type; should be set to "Ethernet" for an Ethernet device.
ONBOOT	Directive that specifies whether the network device is started during the boot process.
BOOTPROTO	May be set to "none" for static configuration or "dhcp" to acquire IP addresses from a DHCP server.
IPADDR0	Static IP address; additional IP addresses can be specified with the variables IPADDR1, IPADDR2, ...
PREFIX	Network mask in CIDR format (i.e., /24)
GATEWAY0	IP address of the default gateway.
DEFROUTE	Binary directive to set the interface as the default route.
DNS1	IP address of the first DNS server.
DOMAIN	Specifies the domain search list in /etc/resolv.conf.
PEERDNS	Binary directive allowing the modification of /etc/resolv.conf.
IPV6INIT	Binary directive that enables the use of IPv6 addressing.
USERCTL	Binary directive to allow users to control a network device.
IPV4_FAILURE_FATAL	Binary directive; if set to "no", when connecting to IPv6 networks, allows the IPv6 configuration to complete if the IPv4 configuration fails.

Indices and tables

- genindex
- modindex
- search

1.2.3 routing

How to enable routing permanently

This file shows if routing is currently enabled on the system (1=on):

```
/proc/sys/net/ipv4/ip_forward
```

To make this change permanent upon reboots change file **/etc/sysctl.conf** to:

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

To implement the changes immediately on the local system, run the following command:

```
# sysctl -p
```

Default Route Configuration

```
# cat /etc/sysconfig/network
NETWORKING=yes
HOSTNAME=testing.example.com
GATEWAY=10.1.6.1
```

Static Routes

```
# ip route show
default via 192.168.122.254 dev ens192 proto static metric 100
192.168.122.0/24 dev ens192 proto kernel scope link src 192.168.122.100 metric 100
192.168.123.0/24 dev virbr2 proto kernel scope link src 192.168.123.1
192.168.124.0/24 dev virbr0 proto kernel scope link src 192.168.124.1
```

```
# ip route add 10.1.5.202/32 via 10.1.7.1 dev eth1
# ip route add 10.1.5.0/24 via 10.1.7.1 dev eth1
```

To make static routes permanent:

```
# vi /etc/sysconfig/network-scripts/route-eth1
```

and append e.g.:

```
10.1.5.0/24 via 10.1.7.1 dev eth1
```

Indices and tables

- genindex
- modindex
- search

1.2.4 dns

HEL 7 includes at least four hostname configuration files of interest: /etc/hostname, /etc/hosts, /etc/resolv.conf, and /etc/nsswitch.conf

/etc/nsswitch.conf

This file defines the order of search for network name entries

```
hosts: files dns
```

Hostname

Change hostname permanently

```
# hostnamectl set-hostname newname
```

Indices and tables

- genindex
- modindex
- search

1.3 Applications

1.3.1 shell

Text Streams and redirection

```
# program 2> error-to-file
# program 2> /dev/null
# progeam &> output-and-error
```

ln - Hardlink vs. Softlink

Hard links are directory entries that point to the same inode. They must be created within the same filesystem. You could delete a hard-linked file in one directory, and it would still exist in the other directory (files are only deleted when the number of dentry records pointing to them hit 0, which is tracked via a counter per file)

On the other hand, a soft link serves as a redirect; when you open a file created with a soft link, the link redirects you to the original file. If you delete the original file, the file is lost. Although the soft link is still there, it has nowhere to go

sed - Stream Editor

Replace all occurences (g - global) of the word Windows with Linux

```
# sed 's/Windows/Linux/g' opsys > newopsys
```

Another example

```
# sed 's/writable = yes/writable = no/g' /etc/samba/smb.conf > ~/smb.conf
```

awk - Aho, Weinberger, and Kernighan

Following command will read out the fourth field in /etc/passwd (the group ID) of every user with a listing of “mike”

```
# awk -F : '/mike/ {print $4}' /etc/passwd
```

Indices and tables

- genindex
- modindex
- search

1.3.2 ssh

ssh tunnel

You can create a ssh tunnel and then connect to the defined port via localhost:

```
ssh -v -C -L 5901:localhost:5901 192.168.100.100
```

This example establishes a ssh tunnel to 192.168.100.100 for port 5901. After session establishment you can vnc to localhost:5901 and it uses the port forwarinding through the tunnel. The option **-L** is the option that makes port forwarding possible

Basic Encrypted Communication

Test *Basic Encrypted Communication*.

Indices and tables

- genindex
- modindex
- search

1.3.3 apache

Installation, Start and Autostart:

```
# yum -y install httpd
# systemctl start httpd
# systemctl enable httpd
# firewall-cmd --permanent --zone=public --add-service http
success
# firewall-cmd --reload
success
```

Copy RHEL CDROM content to webserver directory

Documented under commands-collection yum: *How to create a local repository copy from CDROM / ISO e.g. on web server*.

Indices and tables

- genindex
- modindex
- search

1.3.4 vsftpd

Installation, Start and Autostart:

```
# yum -y install vsftpd
# systemctl start vsftpd
# systemctl enable vsftpd
# firewall-cmd --permanent --zone=public --add-service ftp
success
# firewall-cmd --reload
success
```

Indices and tables

- genindex
- modindex
- search

1.3.5 vncserver

VNC Server Installation auf CentOS 7:

```
https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/system_
↪administrators_guide/ch-tigervnc
https://www.howtoforge.com/vnc-server-installation-on-centos-7
```

Install the vnc server:

```
yum install tigervnc-server
```

@:1 stands for tcp port 5901:

```
cp /lib/systemd/system/vncserver@.service /etc/systemd/system/vncserver@:1.service
vi /etc/systemd/system/vncserver@:1.service
```

Replace <USER> with the actual user whose desktop should be accessed:

```
[...]
[Service]
Type=forking
# Clean any existing files in /tmp/.X11-unix environment
ExecStartPre=/bin/sh -c '/usr/bin/vncserver -kill %i > /dev/null 2>&1 || :'
#ExecStart=/sbin/runuser -l <USER> -c "/usr/bin/vncserver %i"
#PIDFile=/home/<USER>/.vnc/%H%i.pid
ExecStart=/sbin/runuser -l srijan -c "/usr/bin/vncserver %i"
```

```
PIDFile=/home/srijan/.vnc/%H%i.pid
ExecStop=/bin/sh -c '/usr/bin/vncserver -kill %i > /dev/null 2>&1 || :'
```

Adopt firewall rules to allow remote access:

```
firewall-cmd --permanent --zone=public --add-service vnc-server
firewall-cmd --reload
```

Create password for the user (this seems to be saved in cleartext on the system?!) This also starts the server:

```
su - user
vncserver :1
```

Make the service enabled on after every reboot with root credentials **But this destroys the vncserver somehow, could not find out why yet…** So until now I start the vncserver manually with above command after reboots:

```
su -
systemctl daemon-reload

systemctl enable vncserver@:1.service
reboot
systemctl start vncserver@:1.service
```

Indices and tables

- genindex
- modindex
- search

1.3.6 kvm

Mounting installation media for VM installation

This is acutally not a simple task in RH|CentOS.

- CDROM mount passthrough from host to guest is not supported on my system at least…
- Mounting an ISO image brings SELinux issues, qemu cannot access these directories initialls, additional configuration is necessary
- I found the easiest approach is to access the install media via network protocols directly
- In my first successful test I installed vsftpd and copied the CentOS DVD data to /var/ftp/pub/centos and accessed it via virtualization-manager
- **It is most probably a SELinux issue, found a solution for "Permission denied" when mounting .iso images**

  - Had to use one command that allowed me to mount .iso e.g. located in /home/gans/KVM/install: **setsebool -P virt_use_nfs 1**
  - Maybe I also had to use virt-manager as root, but not sure about that

Important Files and directories

- KVM Konfiguration Files: **/etc/libvirt/qemu**
- Disk images: **/var/lib/libvirt/images**

Changes in the XML config files are implemented only after **libvirtd** service restart!

```
# systemctl restart libvirtd
```

Important commands

- **virt-manager**
- **virt-install**
- **virsh**
- **virt-clone**

virt-install

```
# virt-install --name test.example.com \
> --ram=1024 --vcpus=2 \
> --disk path=/var/lib/libvirt/images/test.example.com,size=16 \
> --graphics=spice \
> --location=ftp://192.168.122.100/pub/inst \
> --os-type=linux \
> --os-variant=rhel7
```

If a mistake happens during VM creation, Ctrl+C aborts the process, but the VM will be still running and the same name cannot be reused, because there is a configuration file and virtual disk. To get rid of this wrong VM. . . .

virt-install with kickstart file

For Kickstart installations described later in this chapter, the virt-install command can be used to cite a Kickstart configuration file.

```
--extra-args="ks=ftp://192.168.122.1/pub/ks1.cfg"
```

Example:

```
# virt-install -n outsider1.example.org -r 1024 --disk \
path=/var/lib/libvirt/images/outsider1.example.org.img,size=16 \
-l ftp://192.168.122.1/pub/inst \
-x ks=ftp://192.168.122.1/pub/ks1.cfg
```

Working example from my lab with kickstart file. **Important:** It only worked with an ISO image as install source location !

```
virt-install --name server2.example.com --memory 1024 --vcpus 2 \
--disk /kvm/images/server2.example.com.img,size=16 \
--location /media/iso/CentOS-7-x86_64-Everything-1708.iso --os-type linux --os-
↪variant rhel7 \
--network default --extra-args ks=http://192.168.122.1/inst/ks.cfg
```

Delete a VM from command line

1. Stop the VM:

```
# virsh destroy test.example.com
```

2. Delete associated XML configuration file in **/etc/libvirt/qemu** and virtual disk file in **/var/lib/libvirt/images** (not if it is to be reused)

```
# virsh undefine test.example.com --remove-all-storage
```

3. Now **virt-install** can be run again with same name

virsh

VM management from the command line

- Start VM
- Stop VM
- Delete VM

```
# virsh list --all
# virsh capabilities
# virsh start server1.example.com
# virsh shutdown server1.example.com
# virsh destroy server1.example.com
# virsh autostart server1.example.com
# virsh autostart --disable tester1.example.com
```

If you have to bring down a virbr interface on host like **virbr0** bridge with ich also used by Oracle Virtualbox:

```
# sudo virsh net-destroy default
```

virt-clone

System to be cloned must be shutdown first

```
# virt-clone --original=server1.example.com \
> --name=tester1.example.com \
> --file=/var/lib/libvirt/images/tester1.example.com.img \
> --file=/var/lib/libvirt/images/tester1.example.com-1.img \
> --file=/var/lib/libvirt/images/tester1.example.com-2.img
```

IP addressing and also MAC address seems to be same like in the original image. To scale that kind of additional changes, the **kickstart** feature can be used.

Indices and tables

- genindex
- modindex
- search

1.3.7 List of applications, that should be present on a system

- telnet
- nmap
- mutt - email client
- elinks - web browser
- lftp - ftp client with command completition

1.3.8 man and how to find documentation

whatis and **apropos** can search for man files on different contexts | topics. Info Manuals can be found in **/usr/share/info** Further extensive documentation can be found here **/usr/share/doc** (content depends on application developer effort)

1.4 Commands Collection

1.4.1 dd

dd copy

How to write an iso file to a usb drive:: # dd if=name-of-image.iso of=/dev/sdc bs=512k

Indices and tables

- genindex
- modindex
- search

1.4.2 yum

Find RPM package containing a certain command

```
yum whatprovides */semanage
```

Which packages are installed within the base package group?

Use yum group info:

```
# yum group info base
```

Hint: On the RHEL installation media there is a XML file called **/repodata/*-comps-Server-x86_64.xml** where <grouplist> and <optionlist> shows the belonging package information

How to create a local repository copy from CDROM / ISO e.g. on web server

```
mkdir /var/www/html/inst
chcon -R --reference=/var/www/html /var/www/html/inst
cp -a /media/. /var/www/html/inst/
```

EPEL - Extra Packages for Enterprise Linux

Quite some packages are not included in standard yum repositories on RHEL, Centor, Fedora, ... EPEL is a Fedora Special Interest Group that creates, maintains, and manages a high quality set of additional packages.

```
yum --enablerepo=extras install epel-release
```

How to setup a local yum repository for locally mounted ISO

https://access.redhat.com/solutions/1355683

Issue

- How to set up yum repository to use locally-mounted DVD with Red Hat Enterprise Linux (RHEL) 7
- Would like to upgrade server from RHEL 7.x to RHEL 7.y
- Have a secure environment that will never be connected to the internet, but still needs to be updated
- Way to update the packages on server, with no satellite server and servers disconnected from internet
- Offline patches for Red Hat systems
- How do I create a local repository in RHEL 7?

1. Mount the RHEL 7 installation ISO to a directory like /mnt:

   ```
   # mount -o loop RHEL7.1.iso /mnt
   ```

2. Copy the media.repo file from the root of the mounted directory to /etc/yum.repos.d/ and set the permissions to something sane:

   ```
   # cp /mnt/media.repo /etc/yum.repos.d/rhel7dvd.repo
   # chmod 644 /etc/yum.repos.d/rhel7dvd.repo
   ```

3. Edit the new repo file changing the gpgcheck=0 setting to 1 and adding the following 3 lines:

```
enabled=1
baseurl=file:///mnt/
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release
```

In the end, the new repo file could look like the following:

```
[InstallMedia]
name=DVD for Red Hat Enterprise Linux 7.1 Server
mediaid=1359576196.686790
metadata_expire=-1
gpgcheck=1
cost=500
enabled=1
baseurl=file:///mnt/
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release
```

4. clear the related caches by yum clean all and subscription-manager clean once:

```
# yum clean all
# subscription-manager clean
```

5. check whether you can get the packages list from the DVD repo:

```
# yum  --noplugins list
```

6. if no problem , you wil update:

```
# yum  --noplugins update
```

How to setup a local yum repository for locally mounted CDROM/DVD

http://www.itzgeek.com/how-tos/linux/centos-how-tos/create-local-yum-repository-on-centos-7-rhel-7-using-dvd.html

1. In case of a CDROM, mount the device (cdrom must be connected in esxi):

```
# mkdir /cdrom
# mount /dev/cdrom /cdrom
```

2. Before creating a repo file, move the existing repo files present in /etc/yum.repos.d directory, if not required:

```
mkdir /etc/yum.repos.d/original
mv /etc/yum.repos.d/*.repo /etc/yum.repos.d/original
```

3. Create the new repo file called cdrom.repo under /etc/repos.d directory:

```
# vi /etc/yum.repos.d/local.repo
```

Enter follwing details:

```
[LocalRepo]
name=LocalRepository
baseurl=file:///cdrom
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7
```

4. Clear the repository cache by issuing the following command:

```
# yum clean all
```

Indices and tables

- genindex
- modindex
- search

1.4.3 rpm

Which packages are installed on the system:

```
rpm --query --all
```

In contrast, the following command lists all files in all packages on the local system:

```
# rpm --query -all
```

Indices and tables

- genindex
- modindex
- search

1.4.4 mount

Which formats are supported on the system:

```
[root@centos_server1 ~]# cat /etc/filesystems
 xfs
 ext4
 ext3
 ext2
 nodev proc
 nodev devpts
 iso9660
 vfat
 hfs
 hfsplus
 *
```

Mount ISO File:

```
# mount -o loop rhel-server-7.0-x86_64-dvd.iso /media
```

Technically a loop device is a block device that writes to a file, rather than a piece of hardware. So you always use/need to use the loop back device when mounting a file.

Indices and tables

- genindex
- modindex
- search

1.4.5 ssh

How to do port forwarding via ssh tunnel with ssh: *ssh tunnel*

Indices and tables

- genindex
- modindex
- search

1.4.6 Find Out Linux Distribution And Version

```
# debian cat /etc/issue
Debian GNU/Linux 8 \n \l

# mint cat /etc/issue
Linux Mint 18.2 Sonya \n \l

# redhat cat /etc/issue
\S
Kernel \r on an \m

# redhat cat /etc/redhat-release
CentOS Linux release 7.4.1708 (Core)
```

1.5 Kernel Topics

1.5.1 Verify And Load Kernel Modules

```
# lsmod | grep 8021q
#
# modprobe 8021q
#
# lsmod | grep 8021q
8021q                  33159  0
garp                   14384  1 8021q
mrp                    18542  1 8021q
```

1.5.2 Load Kernel Module Persistent

```
# echo "8021q" > /etc/modules-load.d/8021q.conf
# cat /etc/modules-load.d/8021q.conf
8021q
```

1.6 Security

1.6.1 permissions

chmod

The chmod command uses the numeric value of permissions associated with the owner, group, and others. In Linux, permissions are assigned the following numeric values: r = 4, w = 2, and x = 1. In numerical format, permissions are represented by an octal number, where each digit is associated with a different group of permissions. For example, the permission number 640 means that the owner is assigned permission 6 (read and write), whereas the group has permission 4 (read), and everyone else has no permissions.

Special Permission Bits

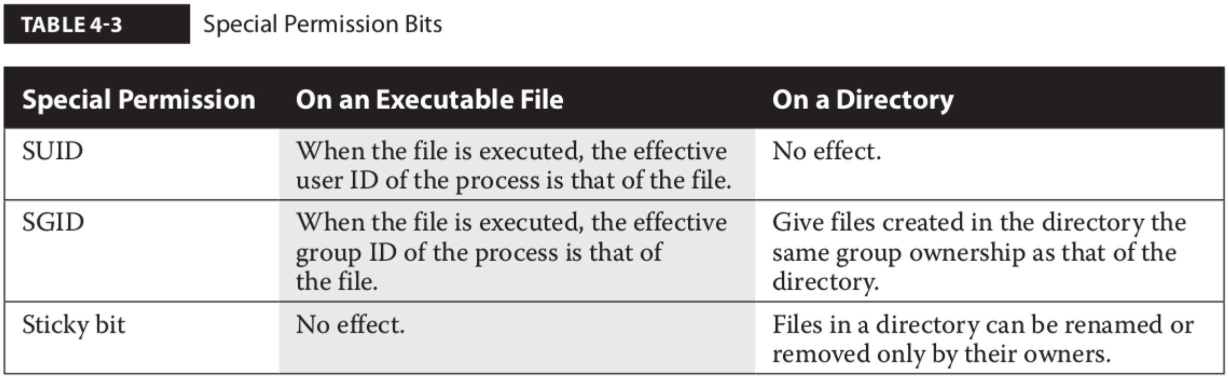


TABLE 4-3 Special Permission Bits

Special Permission	On an Executable File	On a Directory
SUID	When the file is executed, the effective user ID of the process is that of the file.	No effect.
SGID	When the file is executed, the effective group ID of the process is that of the file.	Give files created in the directory the same group ownership as that of the directory.
Sticky bit	No effect.	Files in a directory can be renamed or removed only by their owners.

For the SUID, SGID, and sticky bits, some special options are available. If you choose to use numeric bits, those special bits are assigned numeric values as well, where SUID = 4, SGID = 2, and sticky bit = 1. For example, the following command configures the SUID bit (with the first “4” digit in permission mode). It includes rwx permissions for the user owner (with the “7”), rw permissions for the group owner (with the “6”), and r permission for other users (with the last “4”) on the file named testfile:

```
chmod 4764 testfile
```

If you’d rather use the ugo/rwx format, the following command activates the SGID bit for the local testscript file:

```
chmod g+s testscript
```

And the following command turns on the sticky bit for the /test directory:

```
chmod o+t /test
```

File Attributes

The following command protects /etc/fstab from accidental deletion, even by the root administrative user:

```
chattr +i /etc/fstab
```

With that attribute, if you try to delete the file as the root administrative user, you’ll get the following response:

```
rm /etc/fstab
rm: remove regular file '/etc/fstab'? y
rm: cannot remove '/etc/fstab': Operation not permitted
```

The lsattr command shows an active immutable attribute on /etc/fstab:

```
lsattr /etc/fstab
----i----------- /etc/fstab
```

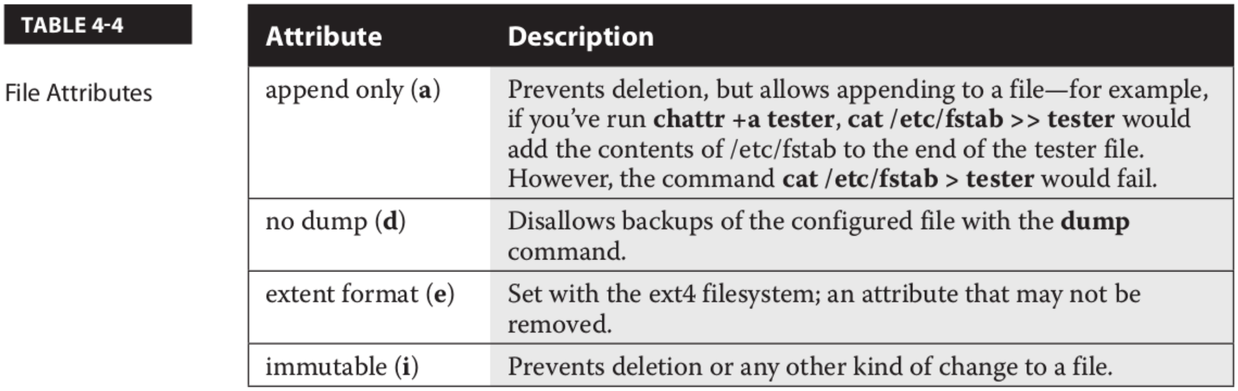


TABLE 4-4

File Attributes

Attribute	Description
append only (**a**)	Prevents deletion, but allows appending to a file—for example, if you've run **chattr +a tester**, **cat /etc/fstab >> tester** would add the contents of /etc/fstab to the end of the tester file. However, the command **cat /etc/fstab > tester** would fail.
no dump (**d**)	Disallows backups of the configured file with the **dump** command.
extent format (**e**)	Set with the ext4 filesystem; an attribute that may not be removed.
immutable (**i**)	Prevents deletion or any other kind of change to a file.

Several key attributes are described in Table 4-4. Other attributes, such as c (compressed), s (secure deletion), and u (undeletable), don't work for files stored in the ext4 and XFS filesystems. The extent format attribute is associated with ext4 systems.

UMASK

Explain Octal umask Mode 022 And 002

If the default settings are not changed, files are created with the access mode 666 and directories with 777. In this example:

- The default umask 002 used for normal user. With this mask default directory permissions are 775 and default file permissions are 664.
- The default umask for the root user is 022 result into default directory permissions are 755 and default file permissions are 644.
- For directories, the base permissions are (rwxrwxrwx) 0777 and for files they are 0666 (rw-rw-rw).

In short:

- A umask of 022 allows only you to write data, but anyone can read data.
- A umask of 077 is good for a completely private system. No other user can read or write your data if umask is set to 077.
- A umask of 002 is good when you share data with other users in the same group. Members of your group can create and modify data files; those outside your group can read data file, but cannot modify it. Set your umask to 007 to completely exclude users who are not group members.

Calculating The Final Permission For FILES

You can simply subtract the umask from the base permissions to determine the final permission for file as follows:

- 666 – 022 = 644

- File base permissions : 666
- umask value : 022
- subtract to get permissions of new file (666-022) : 644 (rw-r–r–)

Default UMASK

With that in mind, the default umask is driven by the /etc/profile and /etc/bashrc files, specifically the following stanza, which drives a value for umask depending on the value of the UID:

```
if [ $UID -gt 199 ] && [ "`id -gn`" = "`id -un`" ]; then
   umask 002
else
   umask 022
fi
```

Access Control Lists

ACLs are supported on ext4 and XFS filesystems, as well as on the Network File System (NFS) version 4.

The getfacl Command

```
[root@server1 ~]# getfacl anaconda-ks.cfg
# file: anaconda-ks.cfg
# owner: root
# group: root
user::rw-
group::---
other::---
```

Make a Filesystem ACL Friendly

RHEL 7 uses the XFS filesystem. When you create an XFS or an ext2/ext3/ext4 filesystem on RHEL 7, ACLs are enabled by default. On the other hand, ext2, ext3, and ext4 filesystems created on older versions of Red Hat may not automatically have ACL support enabled.

To verify whether an ext2/ext3/ext4 filesystem has the acl mount option enabled by default on a partition device such as /dev/sda1, run the command **tune2fs -l/dev/sda1**.

If you want to enable ACL support on a filesystem that does not have the acl mount option configured, you can remount the existing partition appropriately. For example, we can remount the /home partition with ACL using the following command:

```
mount -o remount -o acl /home
```

To make sure this is the way /home is mounted on the next reboot, edit /etc/fstab. Based on the previous command, the associated line might read as follows if /home is formatted with ext4:

```
/dev/sda3   /home   ext4   defaults,acl   1,2
```

Once the change is made to /etc/fstab, you can activate it with the following command:

```
mount -o remount /home
```

To confirm that the /home directory is mounted with the acl option, run the mount command alone, without switches or options. You should see acl in the output, similar to what's shown here:

```
/dev/sda3 on /home type ext4 (rw,acl)
```

Now you can start working with ACL commands to set access control lists on desired files and directories.

Manage ACLs on a File

```
getfacl acltest/testfile
# file: acltest/testfile
# owner: root
# group: root
user::rw-
group::r--
other::r--
```

Set ACL for a user

```
setfacl -m u:michael:rwx acltest/testfile
```

```
getfacl acltest/testfile
# file: acltest/testfile
# owner: root
# group: root
user::rw-
user:michael:rwx
group::r--
mask::rwx
other::r--
```

Set ACL for a group

```
setfacl -m g:users:r acltest/testfile
```

Remove ACLs for a user

```
setfacl -x u:michael acltest/testfile
```

Remove all permissions for a user

```
setfacl -m u:michael:- acltest/testfile
```

Remove all ACLs from a file

```
setfacl -b acltest/testfile
```

```
getfacl acltest/testfile
# file: acltest/testfile
# owner: root
# group: root
user::rw-
group::r--
other::r--
```

One slightly dangerous option relates to other users. For example, the command:

```
setfacl -m o:rwx acltest/testfile
```

Sometimes, you may want to apply such ACLs to all files in a directory. In that case, the **-R** switch can be used to apply changes recursively:

```
setfacl -R -m u:michael:rx /home/gans/acltest
```

Default ACLs

Directories can also contain one or more default ACLs. The concept of a default ACL is similar to a regular ACL entry, with the difference that a default ACL does not have any effect on the current directory permissions, but it is inherited by the files created within the directory.

As an example, if you want all new files and directories in /home/examprep to inherit an ACL that grants read and execute permissions to the user michael, you can run the following command:

```
setfacl -d -m u:michael:rx /home/gans/acltest
```

The -d option in the preceding command specifies that the current operation applies to a default ACL.

ACL and Masks

The mask associated with an ACL limits the permissions available on a file for named users and groups, and for the group owner. The mask shown in Figure 4-2 is rwx, which means there are no limits. If it were set to r, then the only permissions that could be granted with a command such as setfacl is read. To change the mask on the TheAnswers file to read- only, run the following command:

```
setfacl -m mask:r-- /home/gans/acltest/testfile
```

In other words, with a mask of **r–**, you can try to provide other users with all the privileges in the world. But all that can be set with that mask is read privileges.

The mask has an effect only on the group owner and on named users and groups. It does not have any effect on the user owner of the file and on the "other" permission group.

1.6.2 firewall

Firewall status

```
systemctl status firewalld
```

```
iptables -L
```

The iptables tool is the basic foundation that is used by other services to manage system firewall rules. RHEL 7 comes with two such services: the new firewalld daemon and the iptables service, which was included with the previous releases of Red Hat Enterprise Linux. You can interact with firewalld using the graphical utility **firewall-config** or the command- line client **firewall-cmd**.

The iptables and firewalld services both rely on the Netfilter system within the Linux kernel to filter packets. However, whereas **iptables** is based on the concept of "chain of filter rules" to block or forward traffic, firewalld is "zone-based," as you will see in the next sections.

```
iptables -t tabletype <action_direction> <packet_pattern> -j <what_to_do>
```

tabletype

- filter Sets a rule for filtering packets.
- nat Configures network address translation, also known as masquerading

action_direction

- -A (–append) Appends a rule to the end of a chain.
- -D (–delete) Deletes a rule from a chain. Specify the rule by the number or the packet pattern.
- -L (–list) Lists the currently configured rules in the chain.
- -F (–flush) Flushes all the rules in the current iptables chain.

If you're appending to (-A) or deleting from (-D) a chain, you'll want to apply it to network data traveling in one of three directions:

- INPUT All incoming packets are checked against the rules in this chain.
- OUTPUT All outgoing packets are checked against the rules in this chain.
- FORWARD All packets received from a computer and being sent to another computer are checked against the rules in this chain. In other words, these are packets that are routed through the local server.

packet_pattern

- -s ip_address All packets are checked for a specific source IP address.
- -d ip_address All packets are checked for a specific destination IP address.

Packet patterns can be more complex. In TCP/IP, packets are transported using the TCP, UDP, or ICMP protocol. You can specify the protocol with the -p switch, followed by the destination port (–dport). For example, the -p tcp –dport 80 extension affects users outside your network who are trying to make an HTTP connection.

what_to_do

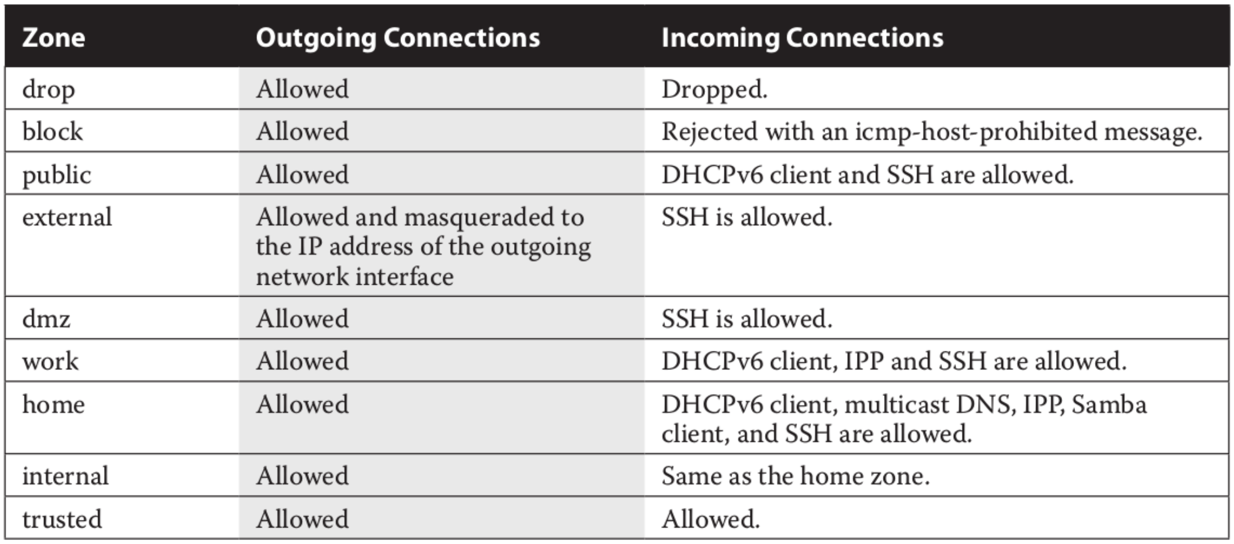
- DROP The packet is dropped. No message is sent to the requesting computer.
- REJECT The packet is dropped. An error message is sent to the requesting computer.
- ACCEPT The packet is allowed to proceed as specified with the -A action: INPUT, OUTPUT, or FORWARD.

The firewalld Service

The firewalld service offers the same functionalities of the iptables tool and more. One of the new features of firewalld is zone-based firewalling. In a zone-based firewall, networks and interfaces are grouped into zones, with each zone configured with a different level of trust. The zones defined in firewalld are listed in Table 4-8, along with their default behavior for outgoing and incoming connections.

A zone is made up of a group of source network addresses and interfaces, plus the rules to process the packets that match those source addresses and network interfaces.

TABLE 4-8 Zones in firewalld



Zone	Outgoing Connections	Incoming Connections
drop	Allowed	Dropped.
block	Allowed	Rejected with an icmp-host-prohibited message.
public	Allowed	DHCPv6 client and SSH are allowed.
external	Allowed and masqueraded to the IP address of the outgoing network interface	SSH is allowed.
dmz	Allowed	SSH is allowed.
work	Allowed	DHCPv6 client, IPP and SSH are allowed.
home	Allowed	DHCPv6 client, multicast DNS, IPP, Samba client, and SSH are allowed.
internal	Allowed	Same as the home zone.
trusted	Allowed	Allowed.

The Zone tab includes all the zones previously listed. When an incoming packet hits the firewall, its source address is checked for a match with the network addresses that belong to the existing zones. If no match is found, the incoming interface of the packet is checked to verify whether it belongs to a zone. Once a correspondence is found, the packet is processed according to the rules of the zone it has been matched to.

If you switch the **firewall-config** tool into Permanent mode, you can add new services or edit existing ones. To accomplish this task, scroll to the bottom of the Services window and click the corresponding icon to remove, add, or edit a service. If desired, you can also configure custom ports for an existing service by clicking the Add or Edit icon

The Console firewall-cmd Configuration Tool

The firewall-cmd configuration tool has the same features and services as the corresponding GUI tool. In fact, both the graphical firewall-config tool and the command interface firewall-cmd are just client front ends that communicate to the underlying firewalld daemon.

As with the GUI tool, **firewall-cmd** can display all the available zones and switch to a different default zone. In the following example, the default zone is changed from the public to the internal zone:

```
# firewall-cmd --get-default-zone
public
# firewall-cmd --set-default-zone=internal
success
# firewall-cmd --get-default-zone
internal
#
```

The option **–list-all** is particularly useful. It lists all the configured interfaces and services allowed through a zone, as illustrated next:

```
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: ens192
  sources:
  services: ssh dhcpv6-client http ftp vnc-server
  ports:
  protocols:
```

```
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

As with many of the firewall-cmd command options, the default zone is assumed if no zone is specified with the –zone command switch. You can add and remove ports and services from a zone with the **–add-port, –add-service, –remove-port, and –remove-service** switches, respectively. The next example enables the http service for traffic hitting the dmz zone:

```
# firewall-cmd --zone=dmz --add-service=http
success
#
```

By default, all configuration changes made by **firewall-cmd** do not survive a server reboot. To make a change that survives a reboot, add the **–permanent** switch to firewall-cmd. Then, run firewall-cmd –reload to implement the change immediately.

Return to the original system. Run the following commands to install and start the telnet service:

```
yum install telnet-server
systemctl start telnet.socket
```

Example Firewall configuration

Run the following command to show the current settings for the default zone:

```
firewall-cmd --list-all
```

Allow telnet traffic through the default zone. Don't forget the –permanent switch to make the change persistent:

```
firewall-cmd --permanent --add-service=telnet
```

Apply the previous change to the run-time configuration of the firewall:

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

1.6.3 encryption

Basic Encrypted Communication

The **ssh-keygen** command is used to set up a public/private key pair. Although it creates an RSA key by default, it also can be used to create a DSA or ECDSA key.

```
ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:498z+yooD85c9e9gotxKe8/CylZK7A2IxJVxd6P7drs root@kvm-server01.example.com
The key's randomart image is:
+---[RSA 2048]----+
|      .o. . o    |
|      o. . o .   |
|   . .    .      |
|    o      .     |
|   . . oS o      |
|    . ..+o.o     |
|      .o+B. * .  |
|     +.*==**o+ . |
|      ++O+o+*BE. |
+----[SHA256]-----+
[root@kvm-server01 gans]#
```

If desired, you can set up RSA keys with a larger number of bits. In our testing, we were able to set up key pairs with up to 8192 bits fairly quickly, even on a virtual machine system with just one virtual CPU.

The command that starts the process is

```
ssh-keygen -b 8192
```

Alternatively, if a DSA key is needed, the following command can help. Only 1024-bit DSA keys are allowed. The process after this command is the same as shown in Figure 4-8.

```
ssh-keygen -t dsa
```

The next step is to transmit the public key to a remote system. It might be one of the servers you administer. If you're willing to transmit that public key over the network (once per connection), the following command can work:

```
ssh-copy-id -i .ssh/id_rsa.pub michael@tester1.example.com
```

1.6.4 selinux

Basic Features of SELinux

The SELinux security model is based on subjects, objects, and actions.

- A subject is a process, such as a running command or an application such as the Apache web server in operation.
- An object is a file, a device, a socket, or in general any resource that can be accessed by a subject.
- An action is what may be done by the subject to the object.

SELinux assigns different **contexts** to objects. A context is just a label, which is used by the SELinux security policy to determine whether a subject's action on an object is allowed or not.

To see the context of a particular file, run the **ls -Z** command.

SELinux Status

As suggested in the RHCSA objectives, you need to know how to "set enforcing and permissive modes for SELinux." There are three available modes for SELinux: **enforcing, permissive, and disabled**. The enforcing and disabled modes are self-explanatory.

```
ls -Z
drwxr-xr-x. root root unconfined_u:object_r:user_home_t:s0 acltest
drwxr-xr-x. gans gans system_u:object_r:user_home_t:s0 Desktop
```

```
drwxr-xr-x. gans gans system_u:object_r:user_home_t:s0 Documents
drwxr-xr-x. gans gans system_u:object_r:user_home_t:s0 Downloads
drwxr-xr-x. gans gans system_u:object_r:audio_home_t:s0 Music
drwxr-xr-x. gans gans system_u:object_r:user_home_t:s0 Pictures
drwxr-xr-x. gans gans system_u:object_r:user_home_t:s0 Public
drwxr-xr-x. gans gans system_u:object_r:user_home_t:s0 Templates
drwxr-xr-x. gans gans system_u:object_r:user_home_t:s0 Videos
```

Equivalent commands for processes, network connections

```
ps -Zaux
ss -Z
netstat -Z
```

SELinux in **permissive** mode means that any SELinux rules that are violated are logged, but the violation does not stop any action. If you want to change the default SELinux mode, change the SELINUX directive in the **/etc/selinux/config** file. The next time you reboot, the changes are applied to the system.

If SELinux is configured in enforcing mode, it protects systems in one of two ways: in targeted mode or in mls mode. The default is the targeted policy, which allows you to customize what is protected by SELinux in a fine-grained manner. In contrast, MLS goes a step further, using the Bell-La Padula model developed for the US Department of Defense.

If you just want to experiment with SELinux, configure it in permissive mode. It'll log any violations without stopping anything. It's easy to set up with the SELinux Administration tool, or you can set SELINUX=permissive in /etc/selinux/config. If the auditd service is running, violations are logged in the audit.log file in the /var/log/audit directory. Just remember, it's likely that Red Hat wants candidates to configure SELinux in enforcing mode during the exams.

Basic SELINUX Configuration and Settings

```
getenforce
Enforcing
```

```
sestatus
SELinux status:                 enabled
SELinuxfs mount:                /sys/fs/selinux
SELinux root directory:         /etc/selinux
Loaded policy name:             targeted
Current mode:                   enforcing
Mode from config file:          enforcing
Policy MLS status:              enabled
Policy deny_unknown status:     allowed
Max kernel policy version:      28
```

You can change the current SELinux status with the setenforce command; the options are straightforward:

```
setenforce enforcing
setenforce permissive
```

To make this change permanent, you'll have to modify the SELINUX variable in the **/etc/selinux/config** file.

Setting Context Types (Essentials for RHCSA !)

There are two commands to set context type:

  - semanage - This is the command you want to use
  - chcon - Use it only in specific cases - **should normally be avoided** - does not survive file system is relabeled !

To set context using semanage, you need to find out the correct context. The easy way is to look for default context settings for alreasy existing items (e.g. files / directories)

```
ls -Z /var/www
drwxr-xr-x. root root system_u:object_r:httpd_sys_script_exec_t:s0 cgi-bin
drwxr-xr-x. root root system_u:object_r:httpd_sys_content_t:s0 html
```

To set this type for html directory to any new directory that should be accessible by the Apache web server, use the following command:

```
semanage fcontext -a -t httpd_sys_content_t "/mydir(/.*)?"
```

The option **-a** is used to add a context type. The option **-t** is used to change a context type (as opposed to user and role)

Setting the context this way is not enough, because it writes it only to the policy and not to the file system. To complete the task, you need to apply the policy to the file system:

```
restorecon -R -v /mydir
```

It is not easy to remember the **semanage fcontext** command. **There is a good man page for it - man semanage-fcontext - with useful examples at the bottom !**

Finding the Context Type You Need

This is one of the challenging parts of setting SELINUX contexts. Roughly, there are three approaches:

- Look at the default environment
- Read the configuration files
- Use **man -k _selinux** to find SELinux-specific man pages for your service

The **man -k _selinux** pages are not installed by default. To install them you need the package **policycoreutils-devel** and then another task:

```
sepolicy manpage -a -p /usr/share/man/man8
```

This is an essential skill - you should master it before going to the exam:

1. **man -k _selinux** - it will only provide one or two man pages
2. **yum whatprovides */sepolicy**
3. **yum install policycoreutils-devel**
4. **sepolicy manpage -a -p /usr/share/man/man8**
5. **man -k _selinux** - you'll see no changes
6. **mandb**
7. This takes some time, after that a lot more man pages should appear with the man command
8. **man -k _selinux | grep http** as example for finding the correct context

Configure Regular Users for SELinux (not needed for RHCSA !)

To review the status of current SELinux users, run the semanage login -l command

```
semanage login -l

Login Name           SELinux User         MLS/MCS Range        Service

__default__          unconfined_u         s0-s0:c0.c1023       *
root                 unconfined_u         s0-s0:c0.c1023       *
system_u             system_u             s0-s0:c0.c1023       *
```

In other words, regular "default" users have the same SELinux user context of the root administrative user. To confirm, run the id -Z command as a regular user. Without changes, it leads to the following output, which suggests that the user is not confined by any SELinux settings:

```
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

In other words, regular "default" users have the same SELinux user context of the root administrative user. To confirm, run the id -Z command as a regular user.

The preceding string defines what is called a label in SELinux jargon. A label is made up of several context strings, separated by a column: a user context (which ends with a _u), a role context (which ends with _r), a type context (which ends with _t), a sensitivity context, and a category set. The rules of the targeted policy, which is the default SELinux policy in RHEL 7, are mostly associated with the type (_t) context.

Although it may not be an exam requirement, regular users should be confined by SELinux. When user accounts are compromised, and they will be compromised, you want any damage that might be caused limited by SELinux rules. The following example specifies a confinement rule that adds (-a) regular user michael, specifying (-s) the user_u context for confinement:

```
semanage login -a -s user_u michael
```

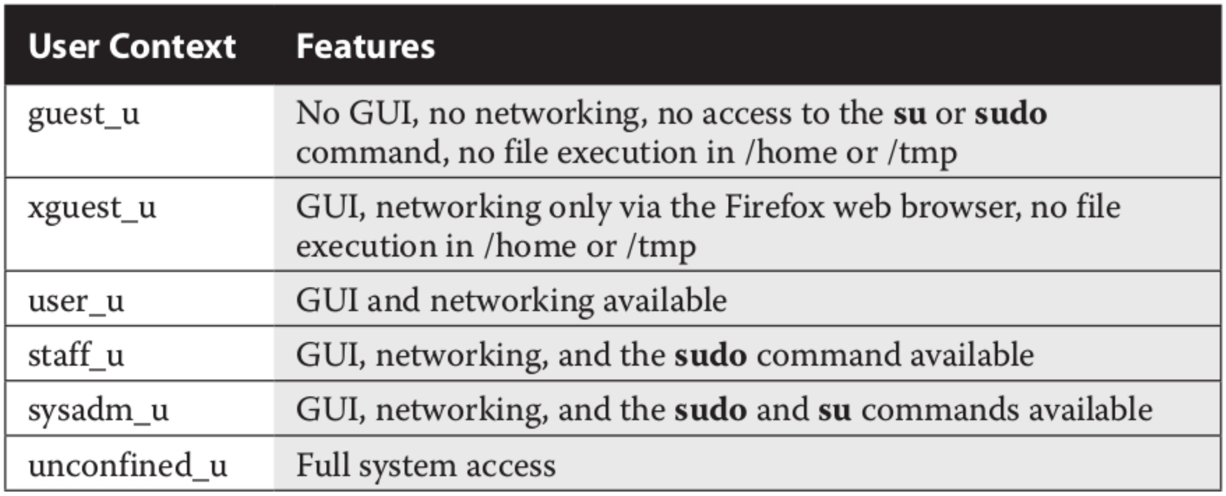
f desired, you can reverse the process with the **semanage -d michael** command.

When a user role is changed, it doesn't take effect until the next login. For example, if we were to change the role for user michael to user_u in a GUI-based command line, the change would not take effect until we logged out and logged back in to the GUI. If you were to try this on your system, you would no longer be able to start any administrative configuration tools, and you would not have access to the sudo and su commands. On some networks, you may want to change the role of future users to user_u. If you don't want regular users tinkering with administrative tools, you could make that change for future default users with the following command:

```
semanage login -m -S targeted -s "user_u" -r s0 __default__
```

This command modifies (-m) the targeted policy store (-S), with SELinux user (-s) user_u, with the MLS s0 range (-r) for the default user. Here, "__default__" includes two underscore characters on each side of the word. As long as user_u is in effect for the default SELinux user, regular users won't have access to use administrative tools or commands such as su and sudo. The following command reverses the process:

```
semanage login -m -S targeted -s "unconfined_u" \
-r s0-s0:c0.c1023 __default__
```



User Context	Features
guest_u	No GUI, no networking, no access to the **su** or **sudo** command, no file execution in /home or /tmp
xguest_u	GUI, networking only via the Firefox web browser, no file execution in /home or /tmp
user_u	GUI and networking available
staff_u	GUI, networking, and the **sudo** command available
sysadm_u	GUI, networking, and the **sudo** and **su** commands available
unconfined_u	Full system access

Manage SELinux Boolean Settings

Most SELinux settings are boolean- in other words, they're activated and deactivated by setting them to 1 or 0, respectively. Once set, the booleans can be retrieved from the /sys/fs/selinux/booleans directory. One simple example is selinuxuser_ping, which is normally set to 1, which allows users to run the ping and traceroute commands.

These settings can be read with the **getsebool -a** and modified with the **setsebool** commands. For example, the following output from the getsebool user_exec_content command confirms that SELinux allows users to execute scripts either in their home directories or from the /tmp directory:

```
user_exec_content --> on
```

For example, the following command disables the noted boolean until the system is rebooted:

```
setsebool user_exec_content off
```

A full list of available booleans is available in the output to the **getsebool -a** command. For more information on each boolean, run the **semanage boolean -l** command.

```
getsebool -a
semanage boolean -l
```

List and Identify SELinux File Contexts

If you've enabled SELinux, the ls -Z command lists current SELinux file contexts, as shown earlier in Figure 4-10. As an example, take the relevant output for the anaconda-ks.cfg file from the /root directory:

```
-rw-------. root root system_u:object_r:admin_home_t:s0 anaconda-ks.cfg
```

It specifies four elements of SELinux security: **the user, role, type, and MLS level** for the noted file. Generally, the SELinux user associated with a file is system_u or unconfined_u, and this generally does not affect access. In most cases, files are associated with an object_r, an object role for the file. It's certainly possible that future versions of the SELinux targeted policy will include more fine-grained options for the user and role.

The key file context is the type, in this case, **admin_home_t**. When you configured FTP and HTTP servers in Chapter 1, you changed the type of the configured directory and the files therein to match the default type of shared files from those services with the chcon command.

For example, to configure a nonstandard directory for an FTP server, make sure the context matches the default FTP directory. Consider the following command:

```
ls -Z /var/ftp/
drwxr-xr-x. root root system_u:object_r:public_content_t pub
```

If you create an /ftp directory as the root user and run the ls -Zd /ftp command, you'll see the contexts associated with the /ftp directory as shown:

```
drwxr-xr-x. root root unconfined_u:object_r:root_t /ftp
```

To change the context, use the chcon command. If there are subdirectories, you'll want to make sure changes are made recursively with the -R switch. In this case, to change the user and type contexts to match /var/ftp, run the following command:

```
chcon -R -u system_u -t public_content_t /ftp
```

If you want to support uploads to your FTP server, you'll have to assign a different type context, specifically public_content_rw_t. That corresponds to the following command:

```
chcon -R -u system_u -t public_content_rw_t /ftp
```

In Chapter 1, you used a different variation on the chcon command. To use that lesson, the following command uses user, role, and context from the /var/ftp directory and applies the changes recursively:

```
chcon -R --reference /var/ftp /ftp
```

Using **restorecon** is the preferred way to change file contexts because it sets the contexts to the values configured in the SELinux policy. The **chcon** command can modify file contexts to any value passed as an argument, but the change may not survive a filesystem relabeling if a context differs from the default value defined in the SELinux policy. Hence, to avoid mistakes, you should modify contexts in the SELinux policy with **semanage fcontext** and use **restorecon** to change file contexts.

Restore SELinux File Contexts

How are context settings applied?

- If a new file is created, it inherits the context settings from the parent directory
- If a file is copied to a directory, this is considered a new file, so it inherits the context settings from the parent directory
- If a file is moved, or copied while keeping its properties (by using **cp -a**), the original file context settings of the file are applied

Especially the latter of these three situations is easily fixed by using restorecon. It is also possible to relabel the entire file system - this could be a good idea.

```
restorecon -Rv /
```

Default contexts are configured in /etc/selinux/targeted/contexts/files/file_contexts. If you make a mistake and want to restore the original SELinux settings for a file, the restorecon command restores those settings based on the file_contexts configuration file. However, the defaults in a directory may vary. For example, the following command (with the -F switch forcing a change to all contexts rather than just the type context) leads to a different set of contexts for the /ftp directory:

```
# restorecon -F /ftp
# ls -Zd /ftp
drwxr-xr-x. root root system_u:object_r:default_t ftp
```

You may also list all default file contexts rules in file_contexts with the **semanage fcontext -l** command.

As an example, a regular expression that matches the /ftp directory and all files in it is given by the following:

```
/ftp(/.*)?
```

Using this regular expression, we can define a SELinux policy rule that assigns to the /ftp directory and all files in it a default type context. This can be done with the **semanage fcontext -a** command. For example, the following command assigns a default type context of public_content_t to the /ftp directory and all the files in it:

```
semanage fcontext -a -t public_content_t '/ftp(/.*)?'
```

Once you have defined a new default policy context for a filesystem path, you can run the restorecon command to set the contexts to the corresponding default policy values.

The following command restores the context recursively (-R) to the public_content_t value defined previously:

```
restorecon -RF /ftp
ls -Zd /ftp
drwxr-xr-x. root root system_u:object_r:public_content_t ftp
```

Identify SELinux Process Contexts

The ps command lists currently running processes. In a SELinux system, there are contexts for each running process. To see those contexts for all processes currently in operation, run the **ps -eZ** command, which lists every (-e) process SELinux context (-Z).

:: ps -eZ system_u:system_r:irqbalance_t:s0 701 ? 00:00:13 irqbalance system_u:system_r:syslogd_t:s0 705 ? 00:00:13 rsyslogd system_u:system_r:alsa_t:s0 707 ? 00:00:00 alsactl system_u:system_r:policykit_t:s0 708 ? 00:00:30 polkitd system_u:system_r:abrt_t:s0-s0:c0.c1023 710 ? 00:00:00 abrtd system_u:system_r:abrt_watch_log_t:s0 712 ? 00:00:00 abrt-watch-log system_u:system_r:modemmanager_t:s0 713 ? 00:00:00 ModemManager system_u:system_r:fsdaemon_t:s0 716 ? 00:00:00 smartd system_u:system_r:rtkit_daemon_t:s0 720 ? 00:00:02 rtkit-daemon system_u:system_r:systemd_logind_t:s0 722 ? 00:00:07 systemd-logind system_u:system_r:lsmd_t:s0 726 ? 00:00:00 lsmd system_u:system_r:avahi_t:s0 727 ? 00:00:00 avahi-daemon system_u:system_r:abrt_watch_log_t:s0 728 ? 00:00:00 abrt-watch-log system_u:system_r:system_dbusd_t:s0-s0:c0.c1023 731 ? 00:00:30 dbus-daemon system_u:system_r:avahi_t:s0 740 ? 00:00:00 avahi-daemon system_u:system_r:chronyd_t:s0 742 ? 00:00:01 chronyd

Diagnose and Address SELinux Policy Violations

If there's a problem, SELinux is running in enforcing mode, and you're sure there are no problems with the target service or application, don't disable SELinux! Red Hat has made it easier to manage and troubleshoot. According to Red Hat, the top two causes of SELinux- related problems are contexts and boolean settings.

Problems with SELinux should be documented in the associated log file, **audit.log**, in the **/var/log/audit** directory.

First, the audit search (ausearch) command can help filter for specific types of problems. For example, the following command lists all SELinux events associated with the use of the sudo command:

```
ausearch -m avc -c sudo
```

In contrast, the **sealert -a /var/log/audit/audit.log** command may provide more clarity.

The GUI SELinux Administration Tool

If you've taken the time to learn SELinux from the command line, this section should be just a review. For many users, the easiest way to change SELinux settings is with the SELinux Administration tool, which you can start with the **system-config-selinux** command

```
yum install policycoreutils-gui
system-config-selinux
```

1.7 Help

Structured Text Help:: http://www.sphinx-doc.org/en/stable/rest.html#rst-primer

1.7.1 Further Help

Ask herwig.gans@gmail.com for help

CHAPTER 2

Indices and tables

- genindex
- modindex
- search