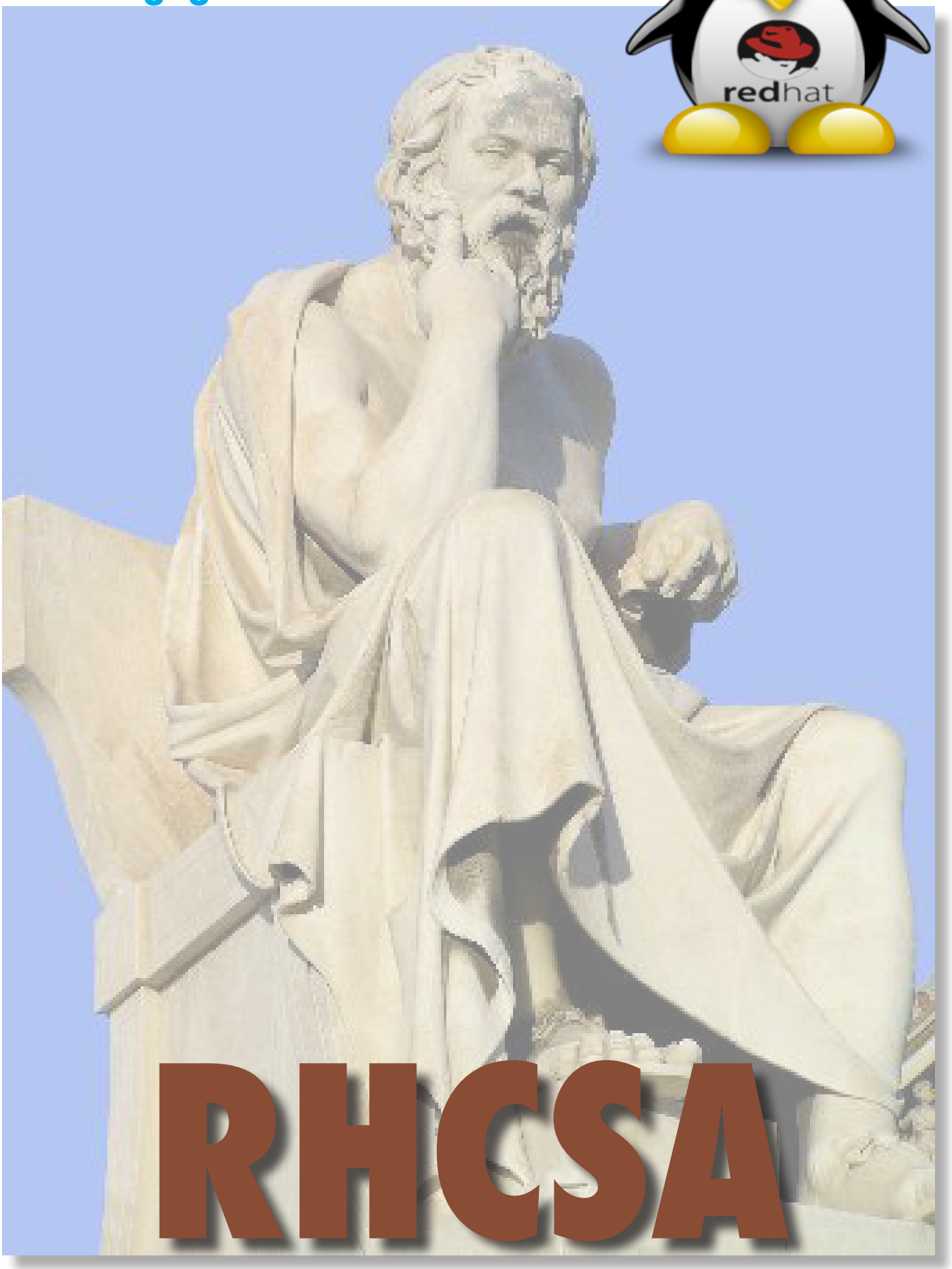
Lemmy's kleiner Prüfungsguide



Inhaltsverzeichnis

Einleitung

Vorwort

Dieses kleine Guide ist das Ergebnis meiner eigenen Prüfungsvorbereitungen für die RedHat-Zertifizierungen RHCSA und RHCE und soll die wichtigsten Prüfungsthemen (siehe Anhang A) soweit zusammanfassen, dass man sich im Idealfall mit seiner Hilfe auf den „RH-300 Rapid-Track Course for Experienced Administrators“ vorbereiten kann.
Aus diesem Grund beschränkt sich dieses Handbuch bei den meisten Themen auf das für die Prüfung notwendige Wissen. Ein für einen späteren Zeitpunkt geplantes Adminhandbuch für die Linux Abteilung wird diese Themen ausführlicher behandeln.

Eine Übersicht der Zertifizierungsinhalte und in welchen Kapiteln ihr die passenden Informationen dazu wiederfindet ist in Anhang A dargestellt.

Ich hoffe, dass euch dieses kleine Guide nützlich sein wird und euch vielleicht sogar hilft die Prüfungen selbst zu bestehen! (:

Michael „Lemmy“ Leimenmeier, 1. Mai 2013

Vorbereitung

Prüfungsrelevante Themen

- Configure a system to run a default configuration HTTP server
- Configure a system to run a default configuration FTP server

Um sich adäquat auf die Prüfungen vorbereiten zu können benötigt jeder Teilnehmer zwei (RHCSA) bzw. drei (RHCE) RedHat Systeme. Da der Themenkomplex auch die Virtualisierung mit KVM einschliesst bietet es sich entsprechend an einen KVM-Host gemeinsam zu installieren und dort entsprechend die benötigten virtuellen Instanzen zu installieren.

Jedem Kursteilnehmer wird eine Nummer zwischen 1 und 9 (folgend mit X dargestellt) zugewiesen anhand derer er später seine UserID, IP Adressen, seine Server etc. zuordnen kann.

Der KVM-Server („*kvmhost*“) wird mit CentOS 6, ScientificLinux 6 oder RHEL 6 installiert, welches OS macht keinen Unterschied da alle drei im Grunde genommen identisch sind; RHEL bietet darüberhinaus lediglich Support und ScientificLinux zusätzliche Pakete, die am CERN benötigt werden.

Der *kvmhost* wird von einem Standard Installationsmedium z.B. einer DVD installiert; es muss sich um ein 64-bit System handeln und über 25GB Storage + 36 GB pro Teilnehme für die virtuellen Instanzen verfügen. SSH, X11 und falls möglich auch VNC-Zugänge sollten in der Firewall freigeschaltet sein.

Nachdem wir den *kvmhost* installiert haben prüfen wir noch die Vorraussetzungen für den Betrieb einer KVM, installieren die benötigten Pakete und richten zu guter Letzt noch einen Apache Webserver, sowie einen vsftpd FTP Server ein und kopieren den Inhalt des Installationsmediums in ein repository Verzeichnis, welches wir via http und ftp zugängig machen um von dort aus mit Kapitel 1 und der Installation der KVM Instanzen beginnen zu können.

Virtuelle Instanzen

Für jeden Teilnehmer werden drei virtuelle Instanzen installiert, für die RHCSA Prüfung benötigen wir nur die ersten beiden Systeme, die beiden im 192.168.122.0/24 Netz liegen. Für die RHCE Zertifizierung benötigen wir noch das dritte System, welches im 192.168.100.0/24 Netz liegen soll um von dort aus Sicherheitstests „von aussen“ durchzuführen.

Virtuelle Systeme auf „kvmhost“

Hostname	Einsatzzweck
serverX.example.com	Workstations und Server die den ganzen Kurs hindurch konfiguriert werden. IP Adresse sollte auf 192.168.122.5X festgelegt werden.
testerX.example.com	Secure Shell Server der remote access unterstützt und Serverdienste zum testen des Clients wie dem Domain Name Service (DNS). Die testerX Systeme bekommen eine IP Adresse von 192.168.122.15X.
outsiderX.example.org	Dieses System wird so konfiguriert, dass es in einem anderen Netz als die anderen beiden Systeme steht und bekommt die IP Adresse 192.168.100.10X. Einige Dienste sollten von diesem System aus nicht erreichbar sein.

Der *kvmhost* ist aus den VMs immer als 192.168.*.1 Gateway zu erreichen.

Partitionierung der Testsysteme

Die virtuellen Systeme sollten jeweils eine virtuelle Festplatte von 12 GB Grösse zugeordnet bekommen. Für den Zweck des Kurses reicht es völlig aus, wenn diese mit einer fixen Partitionierung wie in der Tabelle unten angegeben versehen werden.

Mountpoint	Grösse
/boot	500 MB
/	8 GB
/home	1 GB
swap	1 GB

Paketauswahl

Bei der Installation des *kvmhost* können wir bereits den grundlegenden Einsatzzweck des Servers und somit die Auswahl der Paketgruppen festlegen. Folgende Wahlmöglichkeiten können bei jeder RHEL Installation ausgewählt werden, im konkreten Falle entscheiden wir uns für Virtual Host oder Basic Server.

Kategorie	Beschreibung
Basic Server	Installiert die notwenigen Basispakete um einen RedHat Server zu betreiben
Database Server	Beinhaltet MySQL und PostgreSQL Datenbank Pakete
Web Server	Konfiguriert das System als Apache Webserver
Virtual Host	Konfiguriert das System mit dem KVM VM System
Desktop	Beinhaltet typische Desktopsoftware
Software Development Workstation	Fügt Tools hinzu, die zum modifizieren, kompilieren und debuggen von Software notwendig sind
Minimal	Beinhaltet nur eine minimale Liste zum Betrieb notwendiger Pakete

Die wichtigsten Einzelpakete für die Virtualisierung sind wie folgt:

Paket	Beschreibung
qemu-kvm	Das eigentliche KVM Hauptpaket
python-virtinst	Kommandozeilenwerkzeuge und Libraries um VMs zu erstellen
virt-manager	VM Administrations GUI
virt-top	top Kommando für VM Statistiken
virt-viewer	GUI Verbindung zu den konfigurierten VMs
libvirt	C Toolkit zusammen mit dem libvirtd Service
libvirt-client	C Toolkit für VM Clients

Prüfung des kvmhost

Sind die Virtualisierungspakete alle installiert sollten die nachfolgenden Einstellungen eigentlich alle von den jeweiligen rpm Paketen vorgenommen worden sein. Nichtsdestoweniger prüfen wir die Tauglichkeit des KVM Servers noch einmal im Detail.

Kernel Module:
Ist KVM aktiv sollten mit dem lsmod Kommando folgende Module zu sehen sein:

- kvm
- kvm_intel oder kvm_amd

Prozessorunterstützung:
Die Prozessorflags in der */proc/cpuinfo* Datei sollten *svn* bzw. *vmx* unterstützen.

Virtuelle Netzwerkbrücken:
Für die Verbindung zwischen dem KVM Server und den VMs müssen virtuelle *bridges* konfiguriert sein, über die der Netzwerkverkehr intern geroutet wird. Diese sind mit einem normalen *ifconfig -a* zu sehen, haben wir nur die erste definiert ist das für den RHCSA ausreichend und kann für den RHCE noch nachträglich konfiguriert werden.

```
virbr0 Link encap:Ethernet  HWaddr 9E:56:D5:F3:75:51
       inet addr:192.168.122.1 Bcast:192.168.122.255 Mask:255.255.255.0
virbr1 Link encap:Ethernet  HWaddr 86:23:B8:B8:04:70
       inet addr:192.168.100.1 Bcast:192.168.100.255 Mask:255.255.255.0
```

Kernelparameter für Routingfunktionialität (IP Forwarding)
Im Kernel muss das IP Forwarding aktiviert werden, damit die Pakete der VMs vom *kvmhost* ordnungsgemäß weitergeleitet werden. Um dies bootpersistent zu konfigurieren tragen wir folgenden Wert in die */etc/sysctl.conf*:

```
/etc/sysctl.conf: net.ipv4.ip_forward=1
```

und aktivieren die Änderung entweder über

```
# sysctl -p
```

oder

```
# echo 1 > /proc/sys/net/ipv4/ip_forward
```

Firewalleinstellungen
Die Firewall unter RedHat ist via *iptables* realisiert. Die Konfiguration von iptables wird in */etc/sysconfig/iptables* konfiguriert und kann via *„iptables -L“* abgezeigt werden. Es sollten zwei Einträge vorhanden sein, der erste dient dem Öffnen des SSH Ports (22), der zweite den virtuellen Bridges:

```
-A INPUT -m state --state NEW -m tcp -p tcp --dport 22 -j ACCEPT
-I FORWARD -m physdev --physdev-is-bridged -j ACCEPT
```

SELinux Einstellungen
SELinux sollte im „enforcing“ mode mit einer „targeted“ policy laufen. Unten sehen wir den RHEL6 default, der mit *sestatus* angezeigt werden kann.

```
# sestatus
SELinux status:                 enabled
SELinuxfs mount:                /selinux
Current mode:                   enforcing
Mode from config file:          enforcing
Policy version:                 24
Policy from config file:        targeted
```

Installationsrepository für http- und ftp-Zugriff aufsetzen

In diesem letzten Teil der Einleitung kopieren wir den Inhalt des Installationsmediums, im Beispiel die .iso Datei einer RHEL6.0-DVD auf den *kvmhost* und machen dies via http und ftp für die späteren Installationen der VMs zugänglich. Ganz nebenbei haben wir so die ersten beiden Prüfungsthemen gleich mit abgehakt, nämlich die Einrichtung eines Web- und eines FTP-Servers in der default-Konfiguration (siehe Kästchen zu Beginn des Kapitels).

```
# mkdir -p /kickstart/media/RHEL6.0
# mount -o loop rhel-server-6.0-x86_64-dvd.iso /media
# cp -ar /media/. /kickstart/media/RHEL6.0/
```

Bei‘m Kopiervorgang wählen wir das Quellverzeichnis mit „.“ und nicht „*“ um versteckte Dateien mitzukopieren.

```
# yum install httpd
# /etc/init.d/httpd start
# chkconfig httpd on
```

Wir installieren den apache Webserver via *yum* und sorgen dafür, dass er gestartet ist bzw. nach einem reboot auch ordnungsgemäß wieder gestart wird.

```
# ln -s /kickstart /var/www/html/inst
# chcon -R --reference=/var/www/html/ /var/www/html/inst
# iptables -A INPUT -m state --state NEW -m tcp -p tcp --dport 80 -j ACCEPT
```

Nachdem wir den Inhalt der DVD nach /kickstart/media/RHEL6.0 kopiert haben legen wir im Dokumentenverzeichnis einen Link auf unser Repository Basisverzeichnis an (so können wir auch noch bequem die späteren Kickstart Dateien unter /kickstart ablegen und zugänglich machen). Port 80 muss natürlich auch noch in der lokalen iptables Firewall freigeschaltet werden. Die Zeile in der Mitte ist nötig, da SELinux im strikten „enforcing“ modus läuft und dem apache Prozess den Zugriff auf unser Repository eventuell verbieten würde. Daher ändern wir rekursiv (***-R***) die Dateirechte von /var/www/html/inst bzw. /kickstart anhand der Rechte von /var/www/html, welches wir hier als Referenz nutzen.

```
# yum install vsftpd
# /etc/init.d/vsftpd start
# chkconfig vsftpd on
```

Gleiches Vorgehen wie bei dem Apache Webserver.

```
# ln -s /kickstart /var/ftp/pub/inst
# chcon -R -t public_content_t /var/ftp/pub
# service vsftpd restart
# iptables -A INPUT -m state --state NEW -m tcp -p tcp --dport 21 -j ACCEPT
```

Auch hier verlinken wir einfach in unser /kickstart Verzeichnis, öffnen den Firewall Port 21 für FTP und setzen die Rechte für SELinux, diesmal manuell, indem wir ihm als Typ public_content_t (readonly) direkt mitgeben. Weitere Informationen bzgl. SELinux und Dateirechten finden wir in Kapitel 3.

Virt. Maschinen und automatisierte Installation

Das folgende Kapitel widmet sich drei Themenbereichen. Hauptaufgabenmerk liegt auf der *Virtualisierung* und widmet sich neben den Grundlagen auch den GUI- und CLI-Werkzeugen. Um rücksichts- und hemmungslos auf seinen VMs rumzuwerkeln und auszutesten bietet es sich natürlich an die Installation mit Hilfe von *kickstart* Dateien zu automatisieren, dann kann man seine VMs auch gerne mal zu Oskar in die Tonne legen und solange mit einer anderen weiterarbeiten während sich im Hintergrund die alte neu installiert. Dass man mit dem Austausch des SSH Schlüssels gleich einen Prüfungspunkt mit abhakt sollte man auch nicht unerwähnt lassen.

Prüfungsrelevante Themen

Administration von virtuellen Maschinen

- Access a virtual machines console
- Start and stop virtual machines
- Configure systems to launch virtual machines at boot
- Install RedHat Enterprise Linux as virtual guests

Automatisierte Installation via Kickstart

- Install RedHat Enterprise Linux automatically using Kickstart

Remote Zugriff auf die Server

- Access remote systems using SSH and VNC

KVM, QEMU und libvirt

Die Reihe an Begriffen für die Virtualisierung unter Linux ist auf den ersten Blick recht verwirrend. Während bei RHEL5 noch XEN als Virtualisierungsmittel der Wahl galt hat sich das inzwischen flächendeckend zugunsten von KVM gewandelt und wird von RHEL6 als einzige Virtualisierungsmethode unterstützt. KVM stellt die Kernel Treiber und die Module kvm und je nach Prozessortyp kvm_intel oder kvm_amd zur Verfügung. Darauf baut QEMU (QuickEMUlator) als Hypervisor (Oberaufseher (-:) auf und untersützt neben KVM und XEN noch weitere Virtualisierungen. Jetzt ist das zwar ganz schön virtuelle Prozessoren, Hardwaretreiber und ein Monitoring zu haben, aber ohne Werkzeuge um damit was anzufangen hilft das leider nicht viel.
An diesem Punkt setzt die libvirt mit ihrem libvirtd und den dazugehörigen Werkzeugen an. Die libvirt ist eine VirtualisierungsAPI für C und bietet mit dem libvirtd eine netzwerkfähige Schnittstelle zum QEMU Hypervisor auf der einen und allen Werkzeugen und Konfigurationsdateien und somit letztlich auch zum Admin auf der anderen Seite dar. Sämtliche Kommandos, Python-Skripte und Frameworks wie RHEV setzen auf diese C Schnittstelle auf.

Die Verbindung zu den VMs wird für den Shellzugang über SSH, für die Konsole über die noch einzurichtende serielle ttyS0 Schnittstelle und grafisch ähnlich wie bei einem ilo Board über VNC hergestellt; in der virt-manager GUI ist ein VNC Client bereits implementiert.

Die KVM Kernelmodule haben wir bereits in der Einleitung überprüft, notfalls können sie mit ***modprobe kvm*** geladen werden. Mit Hilfe der libvirt Werkzeuge konnektieren wir uns gegen den QEMU Hypervisor wozu wir den Werkzeugen einen connection String mitgeben müssen. Von der Nutzung der per-user VMs wird aus Performancegründen abgeraten.

QEMU Connection URI Beispiele

URI	Beschreibung
qemu:///session	local access to per-user instance
qemu+unix:///session	local access to per-user instance
qemu:///system	local access to system instance
qemu+unix:///system	local access to system instance
qemu://example.com/system	remote access, TLS/x509
qemu+tcp://example.com/system	remote access, SASl/Kerberos
qemu+ssh://root@example.com/system	remote access, SSH tunnelled

Die Schnittstelle zum Admin stellt wie bereits gesagt die libvirt dar. Dazu verwaltet der libvirtd die Konfigurationen der virtuellen Maschinen und des QEMU Hypervisors über XML-Dateien dar. Diese werden im */etc/libvirt* Verzeichnis gespeichert und beim starten des libvirtd Services in das */var/lib/libvirt* Verzeichnis kopiert um dort die aktuelle Laufzeitkonfiguration darzustellen. Jegliche Änderung in den XML-Dateien in */var/lib/libvirt* sind nach einem restart also verloren, daher entweder unter /etc/libvirt editieren oder aber die CLI-Werkzeuge bzw. die virt-manager GUI nutzen.

libvirt Verzeichnisse

Verzeichnis	Beschreibung
/etc/libvirt	libvirt Konfigurationsdateien, benötigen restart des libvirtd services
/var/lib/libvirt	runtime Konfiguration
/var/lib/libvirt/images	default Verzeichnis für die virtuellen Festplattenimages

libvirt Werkzeuge

Zur Administration stehen uns eine Anzahl von Werkzeugen zur Verfügung, die ich im folgenden kurz vorstellen möchte, bevor wir über die virt-manager GUI zu den Konfigurationsdateien kommen.

LIBVIRT WERKZEUGE

Werkzeug	Beschreibung
virt-manager	Virtual Manager, zentrales GUI Verwaltungstool
virsh	Virtual Shell, zentrales CLI Verwaltungstool
virt-install	Installation via CLI
virt-clone	Clonen von VMs
virt-top	top Auslastung der VMs auf dem Host

Das zentrale Tool zur Verwaltung der virtuellen Maschinen ist die virsh. Wird sie ohne Parameter aufgerufen startet sie in den interaktiven Modus und akzeptiert dort die gleichen Parameter wie direkt auf der Kommandozeile.

VIRSH

Option	Beschreibung
list --all	zeigt alle (auch die inaktiven) VMs an
capabilities	zeigt die Möglichkeiten des Hypervisors an
autostart DOMAIN	setzt das autostart flag der VM damit diese automatisch gestartet wird wenn der Host rebootet wird
edit DOMAIN	öffnet den $EDITOR um die XML Datei der VM direkt zu editieren
start DOMAIN	starten der VM
shutdown DOMAIN	fährt die VM sauber herunter
destroy DOMAIN	klingt gefährlich, ist aber nur ein forced poweroff
undefine DOMAIN	Konfiguration der VM vollständig löschen

```
# virsh list --all
 Id    Name                           State
----------------------------------------------------
 3     server1.example.com            running
```

Für die Installation der VMs kann man entweder den Wizard im virt-manager nutzen oder man nutzt das virt-install Kommando um eine neue virtuelle Maschine entweder von einer editierten XML-Vorlage oder durch die Angabe der Parameter im CLI zu installieren.

VIRT-INSTALL

Option	Beschreibung
--prompt	interaktiver Modus, in dem alles wie im GUI wizard einzeln abgefragt wird
-n | --name	name der VM
-r | --ram	RAM in MB
--disk path=p,size=s	definiert Pfad (p) und Grösse (s) in GB eines Diskimages
-l | --location	url des Installationsrepositories
-x | --extra-args	wird genutzt um zusätzliche Argumente wie Netzwerkkonfiguration, Kickstart-Datei, serielles Terminal etc. mit anzugeben

```
virt-install --name gurke --ram 1024 --vcpus=2,maxvcpus=4 --location=http://192.168.122.1/inst/CentOS6.4/ \
       --os-type=linux --disk path=/var/lib/libvirt/images/gurke.img,size=15 \
       --network network=default,model=virtio --nographics --force \
       --extra-args="console=ttyS0,115200 ks=http://192.168.122.1/inst/ks/ks-standard.cfg ksdevice=eth0
ip=192.168.122.250 netmask=255.255.255.0 gateway=192.168.122.1 dns=192.168.122.1"
```

Damit die Konsole entsprechend genutzt werden kann muss das ttyS0 wie nachfolgend unter „Serielle Konsole einrichten“ konfiguriert und die Optionen --nographics und --extra-args „console=ttyS0,115200“ gesetzt werden.

virt-top ist dem bekannten top Kommando nachempfunden und zeigt den Resourcenverbrauch der virtuellen Maschinen auf dem KVM Server an.

Um eine bereits installierte virtuelle Maschine zu clonen steht einem das virt-clone Kommando zur Verfügung.

VIRT-CLONE

Option	Beschreibung
--prompt	interaktiver Modus, in dem alles wie im GUI wizard einzeln abgefragt wird
-n | --name	name der *neuen* VM
-o | --original	name der *alten* VM
-f | --file	Pfad des *neuen* disk images
--auto-clone	benötigt nur den Namen der *alten* VM (-o) um automatisch zu klonen

Klont den Gast namens demo auf der default connection und generiert den neuen Systemnamen und den Pfad des Diskimages automatisch.

```
# virt-clone --original demo --auto-clone
```

Klont den Gast, der über ein virtuelles Diskimage verfügt

```
# virt-clone --original demo --name newdemo --file /var/lib/xen/images/newdemo.img
```

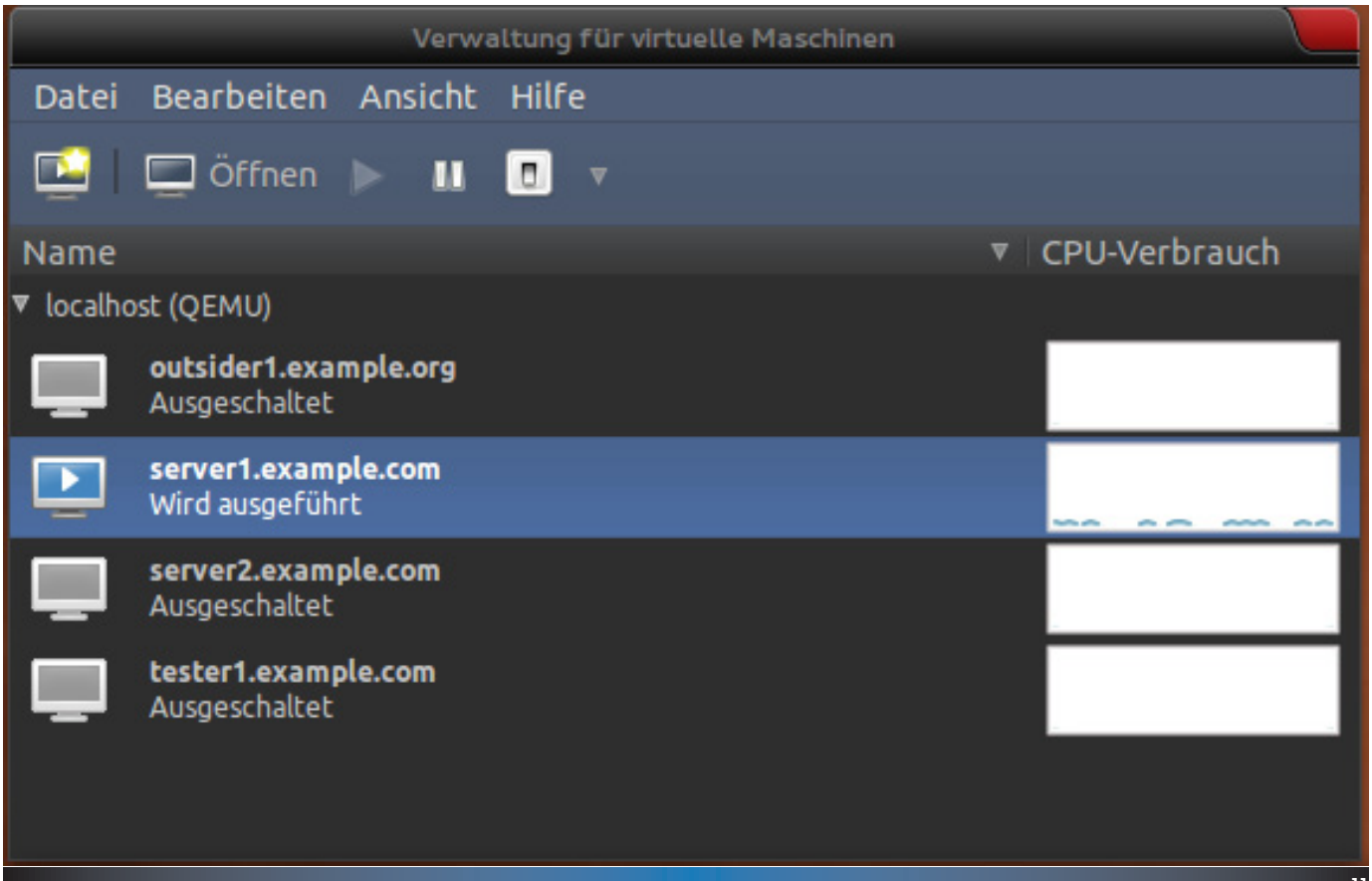
Serielle Konsole einrichten

Um die Konsole der VMs umlenken zu können müssen wir auf dem Host noch das ttyS0 device konfigurieren. Dazu legen wir die Datei */etc/init/ttyS0.conf* mit dem folgenden Inhalt an, bevor wir den Service mit ***start ttyS0*** starten können.

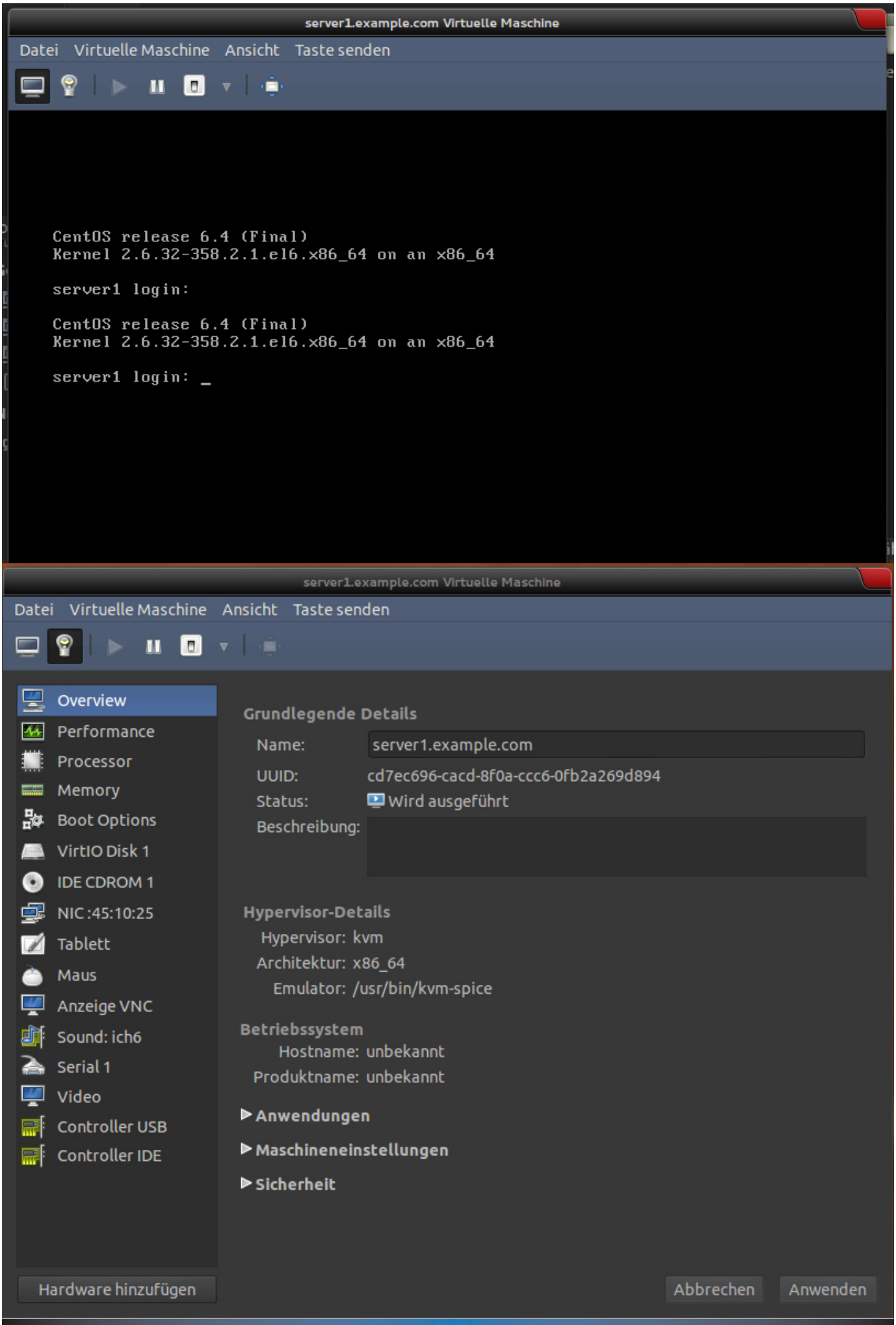
```
# ttyS0 - agetty
stop on runlevel [016]
start on runlevel [345]
instance ttyS0
respawn
pre-start exec /sbin/securetty ttyS0
exec /sbin/agetty /dev/ttyS0 115200 vt100-nav
```

virt-manager

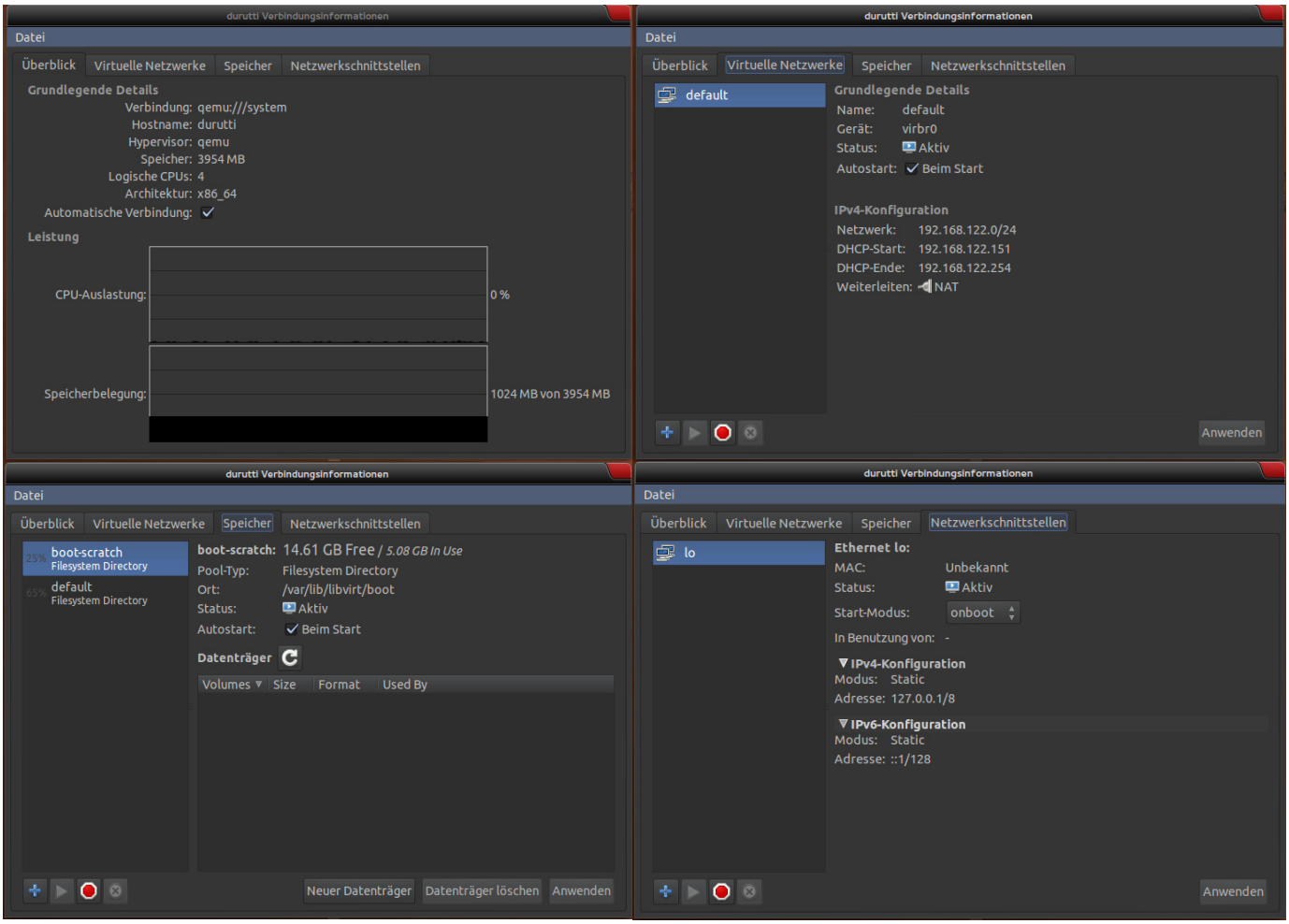
Der virt-manager ist das zentrale GUI Werkzeug für die Administration virtueller Maschinen. Im Hauptfenster findet man eine Übersicht über die konfigurierten und laufenden System und kann von dort aus neue VMs installieren, VMs starten und stoppen, die Konsole der VMs aufrufen oder sogar die VM- und Hypervisor Einstellungen editieren.



Mit einem einfachen Doppelklick auf eine der virtuellen Maschinen öffnet sich das Konsolen- bzw. Detail-Fenster der VM.



Die Einstellung für den QEmu Hypervisor können über den Menüpunkt „Bearbeiten - Verbindungsdetails“ aufgerufen werden und gliedern sich in vier Reiter: Überblick, Virtuelle Netzwerke, Speicher und Netzwerkschnittstellen. Da die GUI grösstenteils selbsterklärend ist zeige ich im folgenden lediglich Screenshots dieser besagten vier Reiter, damit man die Einstellungen wenigstens mal gesehen hat.



Automatisierte Installation via Kickstart Datei

Erzeugen entweder via system-config-kickstart oder kopieren der
Jedes RedHat System bekommt nach der Installation eine kickstart Datei (*/root/anaconda-ks.cfg*) in das Homeverzeichnis des root users abgelegt und spiegelt die Konfiguration des Servers wieder. Diese Datei kann kopiert und als Vorlage für weitere Systeme verwendet werden. Darüberhinaus stellt RedHat natürlich auch ein Konfigurationstool namens ***system-config-kickstart*** für die GUI bereit, in welchem auch bestehende kickstart Dateien geöffnet und bearbeitet werden können. Eine LVM Konfiguration ist aktuell über das GUI tool aber noch nicht möglich.

Aufruf über den bootstring im grub

Damit die kickstart Datei auch genutzt wird muss dem Kernel beim booten diese Position der Datei mit angegeben werden; dies wird wie üblich über den grub boot manager realisiert, in dem man an den boot string z.B. folgende Parameter anhängt:

`ks=hd:sdb1:/ks.cfg`
: Lesen von Festplatte sdb Partition 1

`ks=cdrom:/ks.cfg`
: Lesen von cdrom

`ks=hd:fd0:/ks.cfg`
: Lesen von (virtuellem) Diskettenlaufwerk

`ks=ftp://192.168.122.1/pub/ks.cfg`
: Lesen aus dem pub Verzeichnis eines ftp servers

Kickstart.cfg Parameter

Die kickstart Dateien können mit einem beliebigen ASCII Editor bearbeitet werden; die häufig verwendeten Direktiven werden im Folgenden näher erläutert.

```
install
```
start installation process

Installationsquelle

```
url --url http://192.168.122.1/inst
```
Installationsquelle liegt auf http oder ftp

```
nfs --server 192.168.122.1 --dir=/inst
```
Installationsquelle liegt im NFS

```
harddrive --partition=/dev/sda10 --dir=/home/michael
```
Installation von Platte

```
lang en_US.UTF-8
```
setzen der Systemsprache

```
keyboard us
```
setzen der Tastaturbelegung

Netzwerk

```
network --device eth0 --bootproto static --ip 192.168.122.254
        --netmask 255.255.255.0 --gateway 192.168.122.1
        --nameserver 192.168.122.1 --hostname server.example.com

network --device eth0 --bootproto dhcp
```

```
rootpw --iscrypted $6$...
```
root Passwort setzen

```
firewall --service=ssh
```
Firewall einschalten und Ausnahme für ssh definieren

```
authconfig --enableshadow --passalgo=sha512 --enablefingerprint
```
shadow passport suite konfigurieren

```
selinux --enforcing
```
selinux in enforcing, permissive oder disabled status schalten

```
timezone Europe/Berlin
```
Zeitzone setzen

```
bootloader --location=mbr --driveorder=vda --append="crashkernel=auto rhgb quiet"
```
Konfiguration des bootloaders

```
zerombr yes
```
Deaktiviert die Sicherheitsabfrage ob das Laufwerk formatiert werden soll

```
clearpart --drives=vda --all --initlabel

part /boot --fstype=ext4 --size=500
```
500MB grosse /boot Partition anlegen

```
part swap --size=1024
```
swap partition anlegen

```
part pv.01 --size=1 --grow
```
physical volume auf der Platte anlegen und restlichen Platz allokieren

```
volgroup vg0 pv.01
```
mit der eben erstellen pv.01 Disk eine Volumegruppe vg0 erzeugen

```
logvol / --fstype=ext4 --name=rootvol --vgname=vg0 --size=2048
```
/ mit 2GB als logisches Volume rootvol in der vg0 anlegen

`shutdown, reboot, halt oder poweroff`
: System nach der Installation ausschalten, rebooten etc.

`firstboot --disabled`
: Unterstützt den kickstart Installationsprozess, der der first boot unterbunden wird

Paketauswahl

`%packages`
: Beginn der Paketliste

`@Paket`
: Gruppenpaket Paket installieren (yum groupinstall)

`Paket`
: Einzelnes Paket installieren (yum install)

`-Paket`
: Einzelnes Paket nicht installieren

`%end`
: Ende der Paketliste

`%post`
: Beginn der Postinstall Skripte

Weitere nützliche Tools

Es gibt noch eine Menge weiterer sinnvoller Tools, die zum testen der Netzwerkservices von Nutzen sein können. Ich schlage vor direkt zu Beginn noch folgende Tools zu installieren:

```
# yum install mutt elinks lftp telnet nmap
```

Ports testen mit telnet

```
# telnet localhost 21
```

Offene Ports scannen mit nmap

```
# nmap localhost
```

Mail abrufen/testen mit mutt

```
# mutt -f pop://username@host
```

Webbrowser für die Konsole

```
# elinks http://192.168.122.1/inst
```

FTP shell

```
# lftp ftp.example.org -u username
```

FTP Shell Befehle

Befehl	Beschreibung
cd	Wechselt das aktuelle Verzeichnis auf dem remote host
ls	Listet die Dateien des Kommunikationspartners auf
get	Empfängt eine Datei vom Kommunikationspartner
mget	Empfängt mehrere Dateien (wildcards möglich) vom Kommunikationspartner
put	Sendet eine Datei an den Kommunikationspartner
mput	Sendet mehrere Dateien
pwd	Zeigt den Pfad des aktuellen Verzeichnisses an
quit	Verlässt die ftp shell
lcd	Wechselt das aktuelle Verzeichnis auf dem lokalen Rechner
!ls	Zeigt den Inhalt des aktuellen lokalen Verzeichnisses an
!pwd	Zeigt den Pfad des aktuellen lokalen Verzeichnisses an

Beispielkonfiguration

Im letzten Abschnitt dieses Kapitels möchte ich anhand eines Beispieles die Kickstart Datei, die ich zum aufsetzen des Servers benutzt habe, und die vom ***virt-install*** Kommando bzw. dem Wizard des ***virt-managers*** erzeugte server.xml Datei, die die laufende VM Konfiguration wiederspiegelt, abdrucken.

ks-standard.cfg

```
#platform=x86, AMD64 oder Intel EM64T
#version=DEVEL
# Firewall configuration
firewall --enabled --service=ssh
# Install OS instead of upgrade
install
# Use network installation
url --url="http://192.168.122.1/inst/CentOS6.4/"
# Root password
rootpw --iscrypted $1$oxtL1fmu$MtZsO9s3mqx25iWzfHkLi.
# System authorization information
auth  --useshadow  --passalgo=sha512
# Use text install
text
firstboot --disable
# System keyboard
keyboard de-latin1-nodeadkeys
# System language
lang en_US
# SELinux configuration
selinux --enforcing
# Installation logging level
logging --level=info
# Reboot after installation
reboot
# System timezone
timezone  Europe/Berlin
# Network information
network --bootproto=static --device=eth0 --gateway=192.168.122.1 --ip=192.168.122.250
        --nameserver=192.168.122.1 --netmask=255.255.255.0 --onboot=on
# System bootloader configuration
bootloader --location=mbr
# Clear the Master Boot Record
zerombr
# Partition clearing information
clearpart --all --initlabel
# Disk partitioning information
part /boot --fstype="ext2" --size=512
part pv.01 --size=1 --grow
volgroup vg0 pv.01
logvol swap --name swapvol --vgname=vg0 --size=2048
logvol / --name rootvol --fstype=ext4 --vgname=vg0 --size=10240

%packages
@base
@development
@emacs
@server-platform-devel
@system-admin-tools
@virtualization-client

%end
```

/etc/libvirt/qemu/server1.example.com.xml (von virt-install erzeugte Konfigurationsdatei der VM)

```
<!--
WARNING: THIS IS AN AUTO-GENERATED FILE. CHANGES TO IT ARE LIKELY TO BE
OVERWRITTEN AND LOST. Changes to this xml configuration should be made using:
  virsh edit server1.example.com
or other application using the libvirt API.
-->

<domain type='kvm'>
  <name>server1.example.com</name>
  <uuid>8e994093-65ed-c205-ec34-f8c62e470b89</uuid>
  <memory unit='KiB'>1048576</memory>
  <currentMemory unit='KiB'>1048576</currentMemory>
  <vcpu placement='static' current='2'>4</vcpu>
  <os>
    <type arch='x86_64' machine='rhel6.4.0'>hvm</type>
    <boot dev='hd'/>
  </os>
  <features>
    <acpi/>
    <apic/>
    <pae/>
  </features>
  <clock offset='utc'/>
  <on_poweroff>destroy</on_poweroff>
  <on_reboot>restart</on_reboot>
  <on_crash>restart</on_crash>
  <devices>
    <emulator>/usr/libexec/qemu-kvm</emulator>
    <disk type='file' device='disk'>
      <driver name='qemu' type='raw' cache='none'/>
      <source file='/var/lib/libvirt/images/server1.example.com.img'/>
      <target dev='hda' bus='ide'/>
      <address type='drive' controller='0' bus='0' target='0' unit='0'/>
    </disk>
    <controller type='usb' index='0'>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x01' function='0x2'/>
    </controller>
    <controller type='ide' index='0'>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x01' function='0x1'/>
    </controller>
    <interface type='network'>
      <mac address='52:54:00:47:d0:75'/>
      <source network='default'/>
      <model type='virtio'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x03' function='0x0'/>
    </interface>
    <serial type='pty'>
      <target port='0'/>
    </serial>
    <console type='pty'>
      <target type='serial' port='0'/>
    </console>
    <memballoon model='virtio'>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x04' function='0x0'/>
    </memballoon>
  </devices>
</domain>
```

Grundlagen der Kommandozeile

Dieses Kapitel umfasst zahlreiche Prüfungsthemen rund um die Nutzung der Kommandozeile. Darüberhinaus gibt es an dieser Stelle eine kleine Einführung in die Grundlagen von TCP/IP Netzwerken, die an anderer Stelle in diesem Kursbuch aber noch weiter vertieft werden.

Einige Teilbereiche in diesem Kapitel gehören eigentlich bereits zu den Kerntätigkeiten unseres täglichen Geschäftes, daher habe ich zu den meisten Kommandos auch nicht allzuviel Worte verloren. Nichtsdestotrotz halte ich es für sinnvoll die Kommandos trotzdem hier aufzuführen und sei es nur um sich vor Augen zu halten was in der Prüfung von einem erwartet werden kann.

So kann es zum Beispiel durchaus vorkommen, dass RedHat prüfungsrelevante Informationen im /usr/share/doc Verzeichnis verbirgt, was einen durchaus in die Bredouille bringen kann wenn man da nicht zumindest mal reinschaut (davon ab ist das eh eine gute Idee da mal reinzuschauen, da man dort neben der ausführlichen Dokumentationen zu den installieren Pakete auch noch Beispielkonfigurationen findet, die man schnell nutzen kann um einen Service von Grund auf aufzusetzen).

Prüfungsrelevante Themen

Arbeiten in der Shell
- Access a shell prompt and issue commands with the correct syntax
- Use pipelines and I/O redirection

Pipelines und Umleitungen
- Use input/output redirection (>, >>, |, 2>, etc.)

Datei- und Verzeichnismanagement
- Create/delete/copy/move files and directories
- Create hard- and soft links

Analyse von Textoutput
- Use grep and regular expressions to analyze text output

Lokale Dokumentation
- Locate, read and use system documentation using man, info and files in /usr/share/doc

Nutzung von Texteditoren
- Create and edit text files

Verwaltung von Netzwerkservices
- Start, stop and check the status of network services

Netzwerkkonfiguration und Namensauflösung
- Configure networking and hostname resolution statically or dynamically
- Manage network devices: understand basic IP networking/routing, configure IP addresses/default route statically or dynamically
- Manage name resolution: set local hostname, configure /etc/hosts, configure to use existing DNS server

Textstreams und Kommandos umleiten

`# cat dateiname`
: normale Argumentübergabe

`# database < datafile`
: das programm database bekommt über stdin den Inhalt der Datei datafile geliefert

`# dmesg | less`
: die Standardausgabe stdout von dmesg wird an die Standardeingabe stdin von less geliefert

`# ls > filelist`
: die Ausgabe von ls wird in filelist geschrieben und der vorherige Inhalt von filelist gelöscht

`# ls >> filelist`
: die Ausgabe von ls wird in filelist geschrieben, die Ausgabe wird aber diesmal angehängt

`# program 2> errorlist`
: der Fehlerkanal stderr bzw. 2 wird in die errorlist Datei geschrieben

Standardbefehle

Datei und Verzeichniskonzepte

pwd
gibt das aktuelle Arbeitsverzeichnis aus

Tilde Zeichen (~)
die Tilde steht stellvertretend für das Homeverzeichnis, steht sie alleine ist das eigene Homeverzeichnis, folgt ihr ohne Abstand ein Benutzername z.B. ~mleimenm ist das mit Homeverzeichnis des angegebenen Benutzers gemeint.

Verzeichnispfade
Die Verzeichnispfade unter Unix ähneln dem oberirdischen Teil eines Baumes, dessen Stamm das Wurzelverzeichnis „/“, die Äste die Verzeichnisse und die Blätter die Dateien darstellen. Um in diesem Bild zu bleiben wären die Hardware, Treiber und Mountpoints die unterirdischen Wurzeln, die für den regulären Waldbewohner unsichtbar blieben.
Will man die Position einer Datei in diesem Baum bestimmen gibt es dazu zwei grundlegende Möglichkeiten, entweder man beschreibt den Weg vom Stamm aus, also dem / Verzeichnis, oder aber von seinem aktuellen Standpunkt im Baum aus. Erstere

erkennt man an dem vorangestellten / und nennt man absolute Pfade, bei letzteren fehlt der / natürlich, diese werden relative Pfade genannt.

PATH Umgebungsvariable
Die $PATH Umgebungsvariable dient einem anderen Zweck, und zwar dem Auffinden von Befehlen, die der Benutzer auf der Kommandozeile ausführen möchte. Tippt der Benutzer einen Befehl schaut die shell zuerst in die Liste der eingebauten Kommandos (builtins), wird das Kommando intern nicht gefunden dann zieht die shell die $PATH Variable zu Rate sucht in jedem der dort durch : getrennten Pfade nach dem Befehl und führt diesen aus. Wenn der Nutzer dem Befehl den Pfad voranstellt, z.B. ./ls oder /usr/bin/ls, dann entfällt die Suche natürlich und der konkret benannte Befehl wird bevorzugt.

cd
Das cd Kommando (change directory) dient dem Wechsel des aktuellen Arbeitsverzeichnisses (pwd). Wird kein Pfad angegeben, so springt cd in das Homeverzeichnis des Benutzers zurück.

Dateilisten und ls
Der Inhalt eines Verzeichnisses wird mit dem mächtigen ls Kommando ausgegeben. Je nach Wahl der Schalter werden damit unterschiedliche Dateiattribute angezeigt. Die wichtigsten Optionen sind hierbei: -a um versteckte Dateien mit anzuzeigen, -l für detaillierte Informationen, -t für eine zeitbasierte Auflistung, -i für die Anzeige der inode Nummer und -Z um den SELinux Kontext zu sehen.

Dateien erzeugen und löschen
touch

cp

mv

ln

rm

Verzeichnisse erstellen und löschen
mkdir

rmdir

Umgebungsvariablen
alias
/etc/environment

WILDCARDS

Wildcard	Beschreibung
*	Beliebige Anzahl (auch 0) alphanumerischer Zeichen, z.B. ab* = ab, abc, abd, abe, abcd, usw.
?	Ein einzelnes alphanumerisches Zeichen, z.B. ab? = abc, abd, abe
[]	Eine Reihe von Wahlmöglichkeiten für ein einzelnes alphanumerisches Zeichen, z.B. ab[cd] = abc, abd

Dateien finden
find
locate
/etc/cron.daily/mlocate.cron

Verarbeitung von Textdateien

Lesen von Zeichenströmen (streams)
file

cat

less / more

head / tail

Verarbeiten von Zeichenströmen

sort

grep / egrep

diff

wc

sed

awk

Texte editieren

vi

vipw (-s) / vigr (-s) / visudo
: Editieren der /etc/passwd (/etc/shadow), /etc/group (/etc/gshadow) bzw. /etc/sudoers

emacs

Lokale Onlinedokumentation

Optionsschalter der Kommandos

`-h, --help, --usage`

Manpages

`man [sektion] manpage`

`whatis`
: Sucht nach Begriff im Namen, liefert bei mehreren manpages auch die Sektionen zurück.

`/etc/cron.daily/makewhatis.cron`

`apropos`
: Sucht nach Begriff in der Beschreibung der manpages.

`info`
: Ausführliche infopages, existieren keine infopages wird die passende manpage aufgerufen

Ausführliche Dokumentation

`/usr/share/doc/[Paketname]`

Hier findet man neben ausführlichen Anleitungen zu dem Thema auch viele Beispiele für Konfigurationsdateien. Ausserdem behält sich RedHat vor, hier prüfungsrelevante Informationen zu verstecken!

Einführung in Netzwerkgrundlagen

IPv4 und Adressklassen

Eine IPv4 Adresse (32 bit) besteht aus 4 Gruppen von jeweils einem Byte (dezimal 0-255), die durch Punkte getrennt werden. Die IETF unterteilt 5 verschiedene IPv4 Klassen, die sich in der Relation zwischen der Anzahl der definierbaren Netzwerke und der definierbaren Hosts unterscheidet:

IPv4 Netzwerkklassen

Klasse	Möglicher Adressraum	Beschreibung
A	`1.1.1.1 - 126.255.255.254`	Netzwerke mit bis zu 16 Millionen Hosts
B	`128.0.0.1 - 191.255.255.254`	Netzwerke mit bis zu 65.000 Hosts
C	`192.0.0.1 - 223.255.255.254`	Netzwerke mit bis zu 254 Hosts
D	`224.0.0.1 - 239.255.255.254`	Für Multicast Pakete reserviert
E	`240.0.0.1 - 255.255.255.254`	Reserviert für experimentelle Zwecke

Darin sind Bereiche für private Netzwerke, die nicht direkt mit dem Internet verbunden sind, reserviert: 10.0.0.0, 172.168.0.0 und 192.168.0.0 bis 192.168.255.0.

Um ein IPv4 Netzwerk zu definieren benötigt man darüberhinaus für die Adressierung eine Netzmaske, die den gleichen Aufbau wie die Adresse selbst hat um das jeweilige Netzsegment eindeutig identifizieren zu können. Aus der Kombination der beiden ergibt sich die Netzwerk- und die Broadcast Adresse des Netzsegmentes.

Beispiel:
Eine IP Adresse von 192.168.122.1 und der Standard Netzmaske von 255.255.255.0 bzw. in CIDR-Notation (Classless Interdomain Routing) 192.168.122.0/24 bedeutet eine Netzwerkadresse von 192.168.122.1 und eine Broadcast Adresse von 192.168.122.255. Die 24 setzt sich aus der Anzahl der gesetzten Bits in der Netzmaske zusammen (255.255.255.0 = 11111111.1 1111111.11111111.00000000 = 24 * 1 = /24)

Um die eigene kleine Nachbarschaft zu verlassen benötigt man zu guter Letzt noch ein Tor in die Aussenwelt, Gateway genannt. Während die IP Adresse des Gateways noch innerhalb des lokalen Netzes stehen muss ist dieses Gateway noch mit einem anderen Netz verbunden und schleust die Pakete hindurch. Die IP Adresse des Gateways wird in der Routing Tabelle des lokalen Systems gespeichert und kann via route oder netstat -r angezeigt werden.

IPv6 Adressierung

Eine IPv6 Adresse (128 bit) besteht aus 8 Gruppen von je 4 Hexadezimalzahlen, die durch Doppelpunkte getrennt werden. In IPv6 unterscheidet man grundsätzlich drei verschiedene Adressformate:

- **Unicast**

Eine Unicast Adresse ist mit einem einzelnen Netzwerkadapter verbunden.
Routingfähige Unicast Adressen bestehen aus einem 48bit Netzwerkpräfix, einer 16bit Subnetz ID und einer 64bit ID, die mit der Hardware der NIC verknüpft ist.
Link-Local Unicast Adressen sind lokal und daher nicht routingfähig und setzen sich aus einem 10bit Präfix, gefolgt von 54 Nullen und der gleichen 64bit NIC ID zusammen.

- **Multicast**

Eine Multicast Adresse wird genutzt um eine Nachricht an mehrere Netzwerkadapter gleichzeitig zu schicken. Der Aufbau von Multicast Adressen variiert.

- **Anycast**

Eine Anycast Adresse hat den gleichen grundsätzlichen Aufbau wie eine Unicast Adresse und dienen dazu einen Netzwerkadapter aus einer Liste von möglichen anzusprechen. Dies ist z.B. nützlich, wenn ein Webserver eine RAC Datenbank auf mehreren Nodes ansprechen möchte und es dem client egal ist, welcher dieser Server auf die Anfrage antwortet.

Mit dieser Vielfalt von Adressformaten fallen IPv4 Broadcast Adressen vollständig weg, zu diesem Zweck werden einfach Multicast Adressen verwendet. IPv6 Adressen sind ebenfalls in verschiedene Bereiche unterteilt, die anhand des Präfix unterschieden werden können; manchmal wird die default Adresse auch als ::/128 dargestellt:

IPv6 Adressteile

Präfix / Suffix	Beschreibung
`::1`	Loopback Adresse (Äquivalent zur 127.0.0.1 im IPv4)
`::`	Default Adresse (Äquivalent zur 0.0.0.0 im IPv4)
`fe80::`	Link-Local Adresse, nur eigenes Netzsegment bzw. Point-to-Point Verbindung
`fec0::`	Site-Local Adresse, innerhalb einer Administrationsdomäne (siehe http://en.wikipedia.org/wiki/Unique_local_address)
`ff::`	Multicast Adresse
`2000::`	Globale Unicast Adressen sind routingfähig
`::ffff:0000:0000`	Suffix um eine IPv4 Adresse in eine IPv6 zu manteln (anstelle der Nullen).

IPv6 benutzt eine ähnliches Netzmasken Konzept wie IPv4, die ausschliesslich in CIDR Notation angegeben wird. Das Standard IPv6 Netzwerk hat eine 48bit Netzmaske, wodurch eine Fragmentierung von 16bit Subnetzen möglich ist. Die restlichen 64bit werden für die einzelnen Netzwerkinterfaces verwendet.

Tools, Kommandos und Gateways

ping und ping6
Das ping Kommando wird benutzt um die Netzwerkkonnektivität zu prüfen.

```
ping 127.0.0.1
ping6 -I virbr0 fe80::5652:ff:fe39:24d8
```

Bei'm IPv6 ping6 muss man unter RedHat noch das Interface explizit mit angeben.

ifconfig
Konfiguration der Netzwerkinterfaces.

IFCONFIG

Parameter	Beschreibung
up	Aktiviert den angegebene Adapter
down	Deaktiviert den angegenen Adapter
netmask	Subnetzmaske angeben
broadcast	Broadcast Adresse angeben
metric N	Setzen des metric Wertes für die routing Tabelle des angegebenen Adapters
-arp	Deaktiviert das Address Resolution Protocol (ARP) für den Adapter
promisc	Aktiviert den promiscuous mode des Adapters; in diesem Modus akzeptiert der Adapter auch Pakete die gar nicht für ihn selbst gedacht sind und dient der Netzwerkanalyse oder um Nachrichten zwischen zwei usern mitzuschneiden.
-promisc	Promiscuous mode deaktivieren

Anzeigen der aktuellen Konfiguration

```
# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:50:56:40:1E:6A
          inet addr:192.168.122.50  Bcast:192.168.122.255  Mask:255.255.255.0
          inet6 addr: fe80::2e0:4cff:fee3:d106/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:11253 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1304 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2092656 (1.9 Mb)  TX bytes:161329 (157.5 Kb)
```

Konfiguration des Interfaces

```
# ifconfig eth0 192.168.122.150 netmask 255.255.255.0
```

Aktivieren bzw. deaktivieren des Interfaces

```
# ifconfig eth0 up
# ifconfig eth0 down
```

arp als Diagnosetool
Das ARP Protokoll verknüpft eine IP Adresse mit der Hardware MAC Adresse des Interfaces. Das arp Kommando gibt eine Tabelle mit den bekannten Zuordnungen im lokalen Netzsegment aus. Dies kann z.B. dazu dienen doppelte Adressen von unsauber geklonten Systemen aufzuspüren. Falls nötig kann man mit dem arp Kommando diese Tabellen auch bearbeiten.

```
# arp
Address           HWtype  HWaddress           Flags Mask        Iface
192.168.122.150     ether   52:A5:CB:54:52:A2   C                 eth0
192.168.100.100     ether   00:A0:C5:E2:49:02   C                 eth0
192.168.122.1       ether   00:0E:2E:6D:9E:67   C                 eth0
```

Routing Tabellen mit netstat -r und route
Das netstat Kommando ist das Schweizer Armeemesser unter den Netzwerkkommandos wenn es um die Informationsbeschaffung geht. Es kann die offenen Kanäle für Netzwerkverbindungen, die aktuellen Verbindungen, interface Statistiken und vieles mehr an. An dieser Stelle widmen wir uns aber vorerst nur dem routing, welches man sich mit netstat -r anschauen kann. In vielen Fällen wird noch der -n Schalter genutzt um die Namensauflösung zu umgehen und ausschliesslich IP Adressen anzeigen zu lassen.

```
# netstat -rn
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref Use Iface
192.168.122.0   0.0.0.0         255.255.255.0   U     0      0     0 eth0
0.0.0.0         192.168.122.1   0.0.0.0         UG    0      0     0 eth0
```

Eine Destination von 0.0.0.0 stellt das default Gateway dar. Alle Pakete, für die keine explizit andere Route existiert, wird dort hingeschickt. Liegt das Ziel im lokalen Netzwerksegment wird kein Gateway benötigt und daher entweder 0.0.0.0 oder ein Stern in der Gateway Spalte ausgewiesen. Genmask ist die Netzmaske. Die Routing Flags findet man in dieser Tabelle:

Routing Flags

Flag	Beschreibung
G	Diese Route nutzt ein *Gateway*
U	Der Netzwerkadapter, der in der Iface Spalte angezeigt wird ist aktiv bzw. *up*
H	Über diese Route ist nur ein einziger *Host* zu erreichen
D	Dieser Eintrag wurde durch eine ICMP Redirect Nachricht erstellt
M	Dieser Eintrag wurde durch eine ICMP Redirect Nachricht modifiziert

Während die IPv6 Routing Tabelle komplexer scheint sind die Grundlagen doch die gleichen, mit anderen Worten die IPv6 Gateway Adresse ist mit der default IPv6 Route (::/128) verknüpft. Ausserdem können die gleichen netstat und route Kommandos benutzt werden, wenn sie mit der Option -A inet6 aufgerufen werden.

Dynamische Konfiguration via DHCP
Mit dem dhclient Kommando kann man sich die IP Adresse, die Netzmaske, das default Gateway und die Adresse des zuständigen DNS Servers von einem DHCP Server im Netzwerk zur Verfügung stellen lassen.
dhclient eth0

Netzwerkkonfiguration und Troubleshooting

Konfigurationsdateien

Gibt es Probleme mit dem Netzwerk kann man mit dem Netzwerk Service den aktuellen Status abfragen oder den kompletten Service mit den in den Konfigurationsdateien festgelegten Einstellungen restarten.

```
# service network status
Configured devices:
lo eth0
Currently active devices:
lo eth0 virbr0 vnet0

# service network restart
```

Hilft auch dies nicht, so müssen wir in die Konfigurationsdateien schauen.

/etc/sysconfig/network
```
NETWORKING=yes
HOSTNAME=server1.example.com
GATEWAY=192.168.122.1
```

Steht NETWORKING auf no oder ist der service (chkconfig --list network) gar nicht für den aktuellen runlevel aktiv wird das Netzwerk auch nicht konfiguriert.

/etc/sysconfig/network-scripts/ifcfg-{$INTERFACE}
Im Verzeichnis /etc/sysconfig/network-scripts/ finden wir eine Reihe von ifcfg-* Dateien, die anhand des Interfacenamens unterschieden werden.

```
# cat /etc/sysconfig/network-scripts/ifcfg-eth0
DEVICE=“eth0“
HWADDR=“52:54:00:47:D0:75“
NM_CONTROLLED=“yes“
ONBOOT=“yes“
BOOTPROTO=“static“
TYPE=“Ethernet“
UUID=“0e45caf8-ecae-4386-91a8-844d5ef0bbf4“
IPADDR=“192.168.122.50“
NETMASK=“255.255.255.0“
GATEWAY=“192.168.122.1“
DNS1=“192.168.122.1“
IPV6INIT=“yes“
MTU=“1500“
```

DEVICE und HWADDR legen fest um welches Interface es sich hierbei genau handelt. NM_CONTROLLED legt fest, ob der Interface vom NetworkManager service (service NetworkManager status) konfiguriert wird. Ist ONBOOT gesetzt wird das In-

terface beim booten konfiguriert. Steht BOOTPROTO auf static müssen IPADDR, NETMASK, GATEWAY und DNS1 statisch konfiguriert werden, steht es hingegen auf dhcp, so werden die nötigen Informationen von einem DHCP Server bezogen.

/etc/sysconfig/network-scripts/route-{$INTERFACE}
Hier werden die statischen Routen für das jeweilige $INTERFACE konfiguriert:

```
# cat /etc/sysconfig/network-scripts/route-eth0
ADDRESS0=192.168.100.100
NETMASK0=255.255.255.0
GATEWAY0=192.168.122.1
```

Devicedirektiven für Skripte im /etc/sysconfig/network-scripts/ Verzeichnis

Devicedirektiven für „/etc/sysconfig/network-scripts/“- Skripte

Direktive	Beschreibung
DEVICE	Netzwerkadapter; eth0 ist der erste Ethernet-Adapter
HWADDR	Hardware (MAC) Adresse des Adapters
NM_CONTROLLED	Boolesche Direktive (yes/no), die festlegt ob der Adapter der Kontrolle des NetworkManager Services unterliegt
ONBOOT	Boolesche Direktive, die festlegt ob der Adapter bereits beim booten konfiguriert werden soll
BOOTPROTO	Kann entweder „none“ oder „static“ für eine statische Konfiguration oder „dhcp“ für eine dynamische Konfiguration via DHCP sein
NETMASK	Netzmaske für die statische Konfiguration
TYPE	Netzerktyp, zumeist Ethernet
IPV6INIT	Boolesche Direktive ob IPv6 aktiviert werden soll
USERCTL	Boolesche Direktive ob der reguläre user das Interface konfigurieren darf (bei WLAN oder reinen Applikationsinterfaces sinnvoll)
DEFROUTE	Boolesche Direktive ob das default gateway auch genutzt werden soll
PEERROUTES	Boolesche Direktive, die die Benutzung von definierten Routen erlaubt
IPV4_FAILURE_FATAL	Boolesche Direktive, die wenn ein Fehler auf dem Interface auftritt den Netzwerk Service als failed markiert, so dass dieser über upstart z.B. automatisiert nachgestartet werden kann
NAME	Name des Ethernet devices, falls vorhanden wird damit der default Name des Interfaces ersetzt
UUID	Universal Unique Identifier für diesen Adapter
IPADDR	Statische IP Adresse
GATEWAY	IP Adresse des default gateways

Konfigurationswerkzeuge
Neben den bereits vorgestellen Standardkommandos gibt es von RedHat noch zwei weitere Werkzeuge um die komplette Konfiguration vorzunehmen. Zum einen ist das für die Kommandozeile das ***system-config-network*** und für die Mäuseschubser der ***nm-connection-editor***. Letzterer ist bitte nicht mit dem NetworkManager service zu verwechseln, der vor allem für Desktop User mit variablen Netzwerkzugängen gedacht ist.

Auflösung von Hostnamen
Für die Namensauflösung unter RedHat sind 4 Konfigurationsdateien von Interesse: /etc/sysconfig/network, /etc/nsswitch.conf, /etc/hosts und /etc/resolv.conf. Diese 4 Dateien bestimmen zusammengenommen den lokalen Hostnamen, die lokale Datenbank von Hostnamen und IP Adressen, die IP Adresse eines DNS Servers und die Reihenfolge in welcher diese Datenbanken herangezogen werden.

/etc/nsswitch.conf
In dieser Datei werden Reihenfolgen für zahlreiche Datenbanken definiert. In unserem Falle interessiert uns vorerst nur die Auflösungsreihenfolge der hostnamen:

```
hosts: files dns
```

Er schaut also zuerst in die lokale */etc/hosts* Datei und wenn der den Hostnamen dort nicht findet fragt er beim DNS Server nach.

Veraltete Software kann unter Umständen auch auf die */etc/host.conf* zugreifen.
/etc/host.conf

```
multi on
order hosts,bind
```

/etc/hosts

```
192.168.122.50          server1.example.com
192.168.122.150         tester1.example.com
192.168.100.100         outsider1.example.org
127.0.0.1               localhost.localdomain  localhost
::1                     server1.example.com server1 localhost6.localdomain6 localhost6
```

/etc/resolv.conf

```
search example.com
nameserver 192.168.122.1
```

Die hinter search angegebenen Suchdomänen werden automatisch hinter die hostnamen gehängt falls die einfachen Namen nicht gefunden werden. Darauf folgen ein oder mehrere nameserver Einträge in denen jeweils die IP Adresse eines DNS Servers stehen.

Um zu testen ob der Nameserver ordnungsgemäss arbeitet kann man ihn direkt abfragen:

```
# dig @192.168.122.1 mheducation.com
```

Troubleshooting Tabelle

Lösungsansätze im Problemfall

Problem	Vorgehen
Networking is down.	Check physical connections. Run ifconfig to check active connections. Run the /etc/init.d/network status command. Review the /etc/sysconfig/network file
Unable to access remote systems.	Use the ping command to test access to local, and then remote IP addresses.
Current network settings lead to conflicts.	Check network device configuration in /etc/sysconfig/ network-scripts files. Review settings with the Network Connections tool.
Network settings not consistent.	Check network device configuration in /etc/sysconfig/ network-scripts files. Review settings with the Network Connections tool. The scenario suggests a desire for a static network configuration, so review accordingly.
Hostname is not recognized.	Review /etc/sysconfig/network, run the hostname command, review /etc/hosts for consistency.
Remote hostnames not recognized.	Review /etc/hosts. Check /etc/resolv.conf for an appropriate DNS server IP address. Run the dig command to test the DNS server.

Security auf RHCSA-Niveau

Dieses Kapitel befasst sich mit den Sicherheitsgrundlagen von Linux. Allen voran sind dort natürlich die allseits bekannten owner/group und ugo/rwx Berechtigungen zu nennen, aber kaum befassen wir uns mit den erweiterten Dateiattributen, stellen wir fest, dass sich in den letzten Jahren viel in diesem Bereich getan hat. Von Access Control Listen hat der Eine oder Andere zwar schon mal was gehört aber ernsthaft eingesetzt wird es nur in Firmen, die den Begriff Security zumindest mal periphär gehört haben. Firewalls sind für den Admin des Elisabethanischen Zeitalters leider eine Sache für den Verkehr zwischen Netzwerken, so dass moderneren Bedrohungen natürlich Tür und Tor geöffnet ist. Und zu guter Letzt das Thema SELinux, dass zwar bei RHEL längst in der default Konfiguration standardmässig scharf geschaltet ist, aber von den meisten Administratoren sträflicherweise nicht genutzt, der sich kurz darauf als Vertreter des Australopithecus afarensis stolz den Faustkeil schwingend wundert, dass der chinesische Hacker nebenan bereits seit 2000 Jahren das Schießpulver kennt und mit Gewehren schiesst.

Prüfungsrelevante Themen

Grundlegende Dateiberechtigungen
- list, set, and change standard ugo/rwx permissions

Access Control Listen (ACL)
- Create and manage Access Control Lists (ACLs)

Firewall-Einstellungen
- Configure Firewall Settings using system-config-firewall or iptables

Einführung in Security Enhanced Linux (SELinux)
- Set enforcing/permissive modes for SELinux
- List and Identify SELinux file and process contexts
- Restore default file contexts
- Use boolean settings to modify system SELinux settings

Grundlegende Dateiberechtigungen

Jeder Benutzer eines Unixsystems verfügt über eine userid über die er sich anmeldet und eine primäre sowie evtl. mehrere Sekundärgruppen. Die User- und Gruppennamen, die wir sehen sind in der /etc/passwd bzw. /etc/group numerischen userids und groupids zugeordnet, wobei die ids unter 200 Systemaccounts und die id 0 root vorbehalten sind. Reguläre User beginnen gemäss der Voreinstellungen bei 500.

Analog dazu wird auch jeder Datei und jedem Verzeichnis ein Benutzer (uid) und eine Gruppe (gid) zugewiesen. Auf dieser Basis können unterschiedliche Zugriffsrechte für den Inhaber und die Gruppe der Datei sowie alle restlichen Benutzer vereinbart werden.

Ein ls -l output verdeutlicht das:

```
-rwxr-xr-x.  1 root  root 103432 Aug 13 01:23  /sbin/fdisk
```

Zu Beginn finden wir die Zugriffsrechte (die Datei ist eine normale Datei (-), verfügt über Lese-, Schreib- und Ausführungsrechte für den Inhaber (rwx), sowie Lese- und Ausführrechte für die Mitglieder der angegebenen Gruppe wie auch aller anderen Benutzer (jeweils r-x)). Ausserdem unterliegt die Datei SELinux-Richtlinien (.), hat einen Link count von 1 (existiert also nur einmal auf diesem Filesystem) und gehört dem Benutzer root und der Gruppe root, ist 103432 byte gross und wurde am 13. August um 01:23 Uhr das letzte mal geändert.

Eine Datei kann entweder eine reguläre Datei (-), ein Verzeichnis (*d*), ein block- oder zeichenorientiertes Gerät (*b*/*c*), ein soft link (*l*), Netzwerksocket (*s*) oder eine pipe (*p*) sein.

Reguläre Dateirechte können eine beliebige Kombination aus lesen (*r*), schreiben (*w*) und ausführen (*x*) sein und jeweils für den Inhaber, die Gruppe und alle anderen unabhängig voneinander vergeben werden. Zusätzlich dazu können noch auf den x Schalter das sogenannte *setuid* bzw. *setgid* und das *sticky bit* aufgesetzt werden. Diese werden durch ein s bzw. S (wenn das x-bit darunter selbst nicht gesetzt ist) auf dem x-bit des Inhabers (setuid) oder der Gruppe (setgid) bzw. durch ein t/T auf dem x-bit für alle anderen Benutzer symbolisiert. Ein setuid bzw. setgid bit besagt, dass wenn die Datei ausgeführt wird der Prozess nicht mit den Nutzer- bzw. Gruppenrechten des Nutzers selbst, sondern stattdessen denen der Datei ausgeführt wird; im Falle des oben genannte fdisk könnten dann alle regulären Benutzer mit root Rechten versehen Festplatten partitionieren. Das sticky bit wird unter Linux für Dateien einfach ignoriert und für Verzeichnisse bedeutet es nicht mehr als dass jeder Nutzer dort Dateien erzeugen darf, aber auch nur dieser seine Dateien wieder umbenennen oder löschen darf.

Nur weil eine Datei keinen Schreibzugriff erlaubt heisst das übrigens noch nicht, dass der Benutzer diese auch nicht beschreiben kann, viele Kommandos verfügen über einen -f Schalter um das Überschreiben zu forcieren und auch der vi erlaubt im Gegensatz zu nahezu allen anderen Editoren ein überschreiben mit !.

Darstellung von Dateirechten

Es gibt verschiedene Notationen mit denen diese Dateirechte dargestellt werden können. Wie bekannt kann man diese ausführlich schreiben (*(sst) rwx rwx rwx*) oder in der *ugo/rwx* Notation (z.B. ***chmod ug+rw datei***) oder wahlweise auch in Oktalzahlen

(z.B. ***chmod 0755 datei***).
Wenn wir uns anschauen, dass jede Berechtigung aus drei Schaltern besteht und quasi drei bit belegt sollte der Zusammenhang zwischen der Position der rwx Schalter und der gesetzten bits und der daraus resultierenden Oktalzahlen klar werden (z.B. r-x = 101 = 4 + 0 + 1 = 5 oder nochmal für den Extremfall: rws r-x r-- = s-- rwx r-x r-- = 100 111 101 100 = 4 7 5 4).

umask

Die Rechte, mit denen eine neue Datei angelegt wird, wird durch die sogenannte umask festgelegt. Während diese in früheren Unix-Versionen eine direkt bitmaske darstellte, die bitweise über die Dateirechte gestülpt wurde ist dies heute bei Linux nicht mehr hundertprozentig der Fall.
Für Verzeichnisse verhält es sich zwar noch so, bei Dateien wird aber grundsätzlich kein x-bit mehr automatisch bei der Erzeugung gesetzt.

Eine bitmaske wird sozusagen über die Dateirechte des Ziels gelegt und was nicht durch die Maske gefiltert wurde wird gesetzt (mathematisch betrachtet wird die Maske negiert und dann UND verknüpft): eine Maske von 000 (0) würde zu 111 (7) , 001 (1) zu 110 (6) usw.).

Die Standard umasks unter RedHat sind 002 für uids > 200 und 022 für uids < 200; sprich für reguläre user (uid >= 500) werden neue Dateien mit 664 und neue Verzeichnisse mit 775 erzeugt, für Systemuser (uid < 200) werden Dateien mit 644 und Verzeichnisse mit 755 erzeugt.

Ändern von Dateirechten und -inhabern

Um die Rechte einer Datei zu ändern stehen drei Kommandos zur Verfügung: chmod, chown und chgrp. Alle drei verfügen über einen -R Schalter, mit dem man rekursiv in Verzeichnisse hinabtauchen kann. chmod ändert die Zugriffsrechte der Datei, chown den Inhaber und chgrp die Gruppenzugehörigkeit der Datei, z.B.:

```
# chmod u+x Ch3Lab1
```
gibt dem Inhaber ausführungsrechte auf die Datei

```
# chmod go-w special
```
entzieht der Gruppe und allen anderen Nutzern die Schreibrechte auf die Datei

```
# chmod +x Ch3Lab2
```
gibt Allen (user, group und others) Ausführungsrechte

```
# chmod 4764 testfile
```
setzt die Dateirechte für testfile auf „rws rw- r--“

```
# chmod g+s testscript
```
setzt das setgid bit auf testscript

```
# chmod o+t /test
```
setzt das sticky bit auf das /test Verzeichnis

```
# chown elizabeth F04-01.tif
```
verschenkt die Datei F04-01.tif an die userin elizabeth

```
# chown donna.supervisors F04-01.tif
```
verschenkt die Datei F04-01.tif an die userin donna und die Gruppe supervisors

```
# chgrp project F04-01.tif
```
verschenkt die Datei an die Gruppe project

Spezielle Dateiattribute

Jede Datei unter Linux verfügt neben den Basisattributen auch noch über erweiterte Attribute, die einem dabei helfen können festzulegen was mit den Dateien gemacht werden darf.
Diese Attribute können mit dem lsattr Kommando abgerufen und mit dem chattr Kommando gesetzt werden.

Um dies an einem Beispiel zu zeigen können wir mal eine Datei vor dem versehentlichen verändern / löschen schützen:

```
# chattr +i /etc/fstab

# rm -f /etc/fstab
  rm: cannot remove `/etc/fstab‘: Operation not permitted

# lsattr /etc/fstab
  ----i---------e- /etc/fstab
```

```
# chattr -i /etc/fstab
```

Das ext4 Filesystem unterstützt nicht alle Attribute wie z.B. compressed (c), secure deletion (s) oder undeletable (u), aber die Attribute der folgenden Liste sind nutzbar:

Erweiterte Dateiattribute

Attribut	Beschreibung
append only (a)	Datei darf nicht gelöscht oder überschrieben werden, es dürfen aber Daten in die Datei angehängt werden. Sehr nützlich für authorization logfiles.
no dump (d)	Datei darf nicht via dump command gesichert werden. Ist sinnvoll bei swap Dateien etc.
extent format (e)	Wird vom ext4 Filesystem gesetzt und kann nicht entfernt werden
immutable (i)	Datei kann nicht gelöscht oder geändert werden
indexed (I)	Wird auf Verzeichnisse gesetzt, die mit hash-Bäumen indiziert werden; attribut kann nicht entfernt werden.

Access Control Listen (ACL)

Was aber passiert, wenn z.B. ein logfile von der einen Gruppe erzeugt wird, aber von einer ganz anderen gelesen werden muss, ohne dass man die Datei für die ganze Welt öffnet? Man denke z.B. an das Fraud Management, welches lesend auf Produktionsdaten zugreifen muss ohne dass Dritte an diese Daten kämen. Hierbei helfen sogenannte Zugriffskontrolllisten oder Access Control Lists (kurz ACL) weiter, unter der Vorraussetzung, dass das Filesystem auch mit der acl option gemountet wurde. ACLs erweitern die bestehenden Zugriffsrechte, so dass grundsätzlich jede Datei auch über ACLs per se verfügt:

```
# ls -l CentOS-6.4-x86_64-bin-DVD1.iso
-rw-rw-r--. 1 mleimenm mleimenm 4353378304 Apr 26 19:49 CentOS-6.4-x86_64-bin-DVD1.iso

# getfacl CentOS-6.4-x86_64-bin-DVD1.iso
# file: CentOS-6.4-x86_64-bin-DVD1.iso
# owner: mleimenm
# group: mleimenm
user::rw-
group::rw-
other::r--
```

ACL Einträge haben ein dreigeteiltes Format, welches durch Doppelpunkte getrennt wird. Feld 1 ist entweder user, group, other oder mask, in Feld 2 steht der user- oder gruppenname, ist Feld 2 leer so handelt es sich um den Eigentümer der Datei bzw. die der Datei zugehörige Gruppe und Feld drei verfügt über die bekannten rwx Flags.

Mit getfacl lässt man sich die ACLs einer Datei anzeigen, mit setfacl werden die Einträge bearbeitet :

setfacl

Schalter	Beschreibung
-b | --remove-all	Entfernt alle ACLs von der Datei, behält ugo/rwx Berechtigungen bei
-k	löscht die default ACL Einträge
-m	modifiziert die ACLs einer Datei, üblicherweise unter Angabe von user (u) oder group (g)
-n | --mask	Bezieht die Maske nicht in die Kalkulation der effektiven Zugriffsrechte mit ein
-R	rekursives ändern der ACLs
-x	entfernt einen spezifischen ACL Eintrag

z.B.

```
# setfacl -m g:teachers:r-- /home/examprep/TheAnswers
```

Gibt der Gruppe teachers zusätzlich Leserechte auf /home/examprep/TheAnswers

```
# setfacl -b /home/examprep/TheAnswers
```

Löscht alle ACL Einträge der Datei /home/examprep/TheAnswers

```
# setfacl -m o:--- /home/examprep/TheAnswers
```

Entzieht allen anderen Benutzern (other) sämtliche Zugriffsrechte.

NFS Shares und ACLs

In NFSv4 wurden auch ACLs für NFS Shares eingeführt und erlauben eine feinere Kontrolle der Rechte via nfs4_getfacl und nfs4_setfacl. NFS ACLs haben das folgende Format:

type:flags:principal:permissions

Type ist entweder Allow (A) oder Deny (D), principal kann ein regulärer user oder gruppe sein (kleingeschrieben) oder OWNER, GROUP oder EVERYONE (grossgeschrieben) und verweist auf den Eigentümer oder die zur Datei gehörenden Gruppe bzw. others. Abhängig davon ob es sich bei dem Ziel um ein Verzeichnis oder eine Datei handelt variieren die folgenden permissions leicht und bieten deutlich filigranere Möglichkeit als die regulären rwx bits:

ACL Berechtigungen in NFSv4

Berechtigungsflag	Beschreibung
r	Datei lesen oder Verzeichnis auflisten
w	Datei beschreiben oder Datei im Verzeichnis neu erstellen
a	Daten an eine Datei anfügen oder ein Unterverzeichnis erstellen
x	Datei ausführen oder in ein Verzeichnis wechseln
d	Datei oder Verzeichnis löschen
D	Unterverzeichnis löschen
t	Datei- oder Verzeichnisattribute lesen
T	Datei- oder Verzeichnisattribute schreiben
c	ACLs einer Datei oder eines Verzeichnisses lesen
C	ACLs einer Datei oder eines Verzeichnisses schreiben
y	Datei oder Verzeichnis synchronisieren (NFS Dropbox-Style (-:)

Firewall-Einstellungen

Hier werden grundlegende Firewall Einstellungen via iptables / ip6tables bzgl. des Paketfilterings besprochen die für dei RHCSA Prüfung relevant sind. Weiterführende Themen werden in den späteren RHCE Kapiteln abgehandelt.

iptables basiert auf sogenannten Regelketten, die sequentiell strukturiert sind. Jede der darin enthaltenen Regeln prüft ob das angegebene Kriterium erfüllt ist und definiert eine Aktion, die ausgeführt werden soll, wenn die Kriterien zutreffen. Das iptables Kommando hat das folgende Format:

```
iptables -t tabletype <action direction> <packet pattern> -j <what to do>
```

tabletype kann entweder filter (Paketfilter) oder nat (NAT bzw. masquerading) lauten, wird kein tabletype angegeben wird filter als default angenommen. Danach folgt die Aktionsrichtung (<action direction>), von denen es 4 Basisaktionen gibt:

-A (--append)	hängt eine Regeln an das Ende der Kette
-D (--delete)	löscht eine Regel aus der Kette, die entweder durch die Regelnummer oder wahlweise durch das Paketmuster (<packet pattern>) spezifiziert wird
-L (--list)	zeigt die aktuelle Konfiguration an
-F (--flush)	Setzt die Regeln der aktuellen Kette zurück

Wenn man eine Regel hinzufügen oder löschen möchte dann möchte man üblicherweise angeben welche Pakete davon betroffen sein sollen, dazu dienen diese drei Ketten:

INPUT	Alle für diesen Rechner eintreffenden Pakete
OUTPUT	Alle von diesem Rechner ausgehenden Pakete
FORWARD	Alle von diesem Rechner weitergeleiteten Pakete

Als nächstes muss ein Paketmuster (<packet pattern>) definiert werden. Diese Paketmuster können in ihrer Form beliebig komplex werden, die einfachsten sind die aus IP Adressen:

-s ip_address	Pakete anhand der source Adresse filtern
-d ip_address	Pakete anhand der destination Adresse filtern

Ausserdem kann man noch das Protokoll und den Netzwerkport angeben:

-p protocol	TCP, UDP oder ICMP
--dport port	Portnummer des Services siehe /etc/services

Zu guter Letzt fehlt noch die auszuführende Aktion (<what to do>) wenn das Paketmuster zutrifft, hierzu gibt es drei grundsätzliche Möglichkeiten:

-j DROP	Das Paket wird stillschweigend verworfen, der Sender wird nicht darüber informiert.
-j DENY	Das Paket wird verworfen, der Sender aber über die Ablehnung informiert.
-j ACCEPT	Das Paket darf entsprechend der angegebenen -A Aktion weiterlaufen: INPUT, OUTPUT oder FORWARD

Die aktuelle Konfiguration können wir uns wie gesagt mit iptables -L anschauen, folgend mal die default Konfiguration von RHEL6, in der lediglich der ssh Service erlaubt ist:

```
# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere            state RELATED,ESTABLISHED
ACCEPT     icmp --  anywhere             anywhere
ACCEPT     all  --  anywhere             anywhere
ACCEPT     tcp  --  anywhere             anywhere            state NEW tcp dpt:ssh
REJECT     all  --  anywhere             anywhere            reject-with icmp-host-prohibited

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
REJECT     all  --  anywhere             anywhere            reject-with icmp-host-prohibited

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Die Ketten für IPv6 sehen gar nicht viel anders aus:
# ip6tables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all      anywhere             anywhere            state RELATED,ESTABLISHED
ACCEPT     ipv6-icmp    anywhere             anywhere
ACCEPT     all      anywhere             anywhere
ACCEPT     tcp      anywhere             anywhere            state NEW tcp dpt:ssh
REJECT     all      anywhere             anywhere            reject-with icmp6-adm-prohibited

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
REJECT     all      anywhere             anywhere            reject-with icmp6-adm-prohibited

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

Die Konfiguration wird in /etc/sysconfig/iptables bzw. /etc/sysconfig/ip6tables gespeichert:

```
# cat /etc/sysconfig/iptables
# Firewall configuration written by system-config-firewall
# Manual customization of this file is not recommended.
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
-A INPUT -p icmp -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 22 -j ACCEPT
-A INPUT -j REJECT --reject-with icmp-host-prohibited
-A FORWARD -j REJECT --reject-with icmp-host-prohibited
COMMIT

# cat /etc/sysconfig/ip6tables
# Firewall configuration written by system-config-firewall
# Manual customization of this file is not recommended.
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
-A INPUT -p ipv6-icmp -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 22 -j ACCEPT
-A INPUT -j REJECT --reject-with icmp6-adm-prohibited
-A FORWARD -j REJECT --reject-with icmp6-adm-prohibited
COMMIT
```

*filter zeigt an, dass es sich bei den folgenden Regeln um filter Regeln handelt.

Die nächsten 3 Zeilen legen die default Aktion der jeweiligen Ketten fest, bei RedHat stehen sie per default auf ACCEPT, Sicherheitsspezialisten der US National Security Agency (NSA) empfehlen hingegen für die INPUT und OUTPUT Ketten auf DROP zu setzen. Die [0:0] sind byte und paket zähler.
Die folgenden Zeilen werden eins zu eins an das iptables bzw. ip6tables Kommando durchgereicht, gehen wir mal die unbekannten Optionen durch.

```
-A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
```

ESTABLISHED bedeutet, dass bestehende Verbindungen weiterlaufen dürfen, RELATED bedeutet, dass auch anschliessende Verbindungen (nächste Datei bei ftp Transfer zum Beispiel) erlaubt sind.

```
-A INPUT -p icmp -j ACCEPT
```

Eingehende ICMP Verbindungen werden erlaubt.

```
-A INPUT -i lo -j ACCEPT
```

Eingehende Verbindungen, die über das loopback Interface werden akzeptiert.

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

Diese Zeile ist die einzige in der ganzen Konfiguration, die wirklich neue eingehende reguläre Verbindungen erlaubt. Mit -m legt man fest welches Kriterium „matchen“ (also zutreffen soll), in diesem Falle der „package state“, der mit --state NEW auf neue Verbindungen konkretisiert wird. Ausserdem muss noch mit -m tcp ein weiteres Kriterium erfüllt sein, nämlich eine tcp Verbindung auf port 22 (ssh). Treffen alle diese Kriterien zu wird die Verbindung mit -j ACCEPT akzeptiert.

```
-A INPUT -j REJECT --reject-with icmp-host-prohibited
-A FORWARD -j REJECT --reject-with icmp-host-prohibited
```

Diese beiden Zeilen lehnen alle weiteren (im Falle der FORWARD Kette somit alle) Verbindungen ab und schicken mit --reject-with eine icmp-host-prohibited Meldung an den Sender zurück.

```
COMMIT
```

Commit schliesst die Konfiguration ab und aktiviert die neuen Regeln.

Firewall Konfigurationswerkzeuge

Neben dem dem iptables Kommando oder dem editieren der Konfigurationsdatei stehen einem noch ein Konsolen und GUI Tools zur Verfügung: ***system-config-firewall***, die eigentlich selbst erklärend sind. Setzt man ein solches Tool ein sollte man vorher ein Backup der */etc/sysconfig/ip*tables* Dateien machen, da diese komplett neu geschrieben werden.

Gebräuchliche Services und deren Ports

Service	Beschreibung
Amanda Backup Client	A client associated with the Advanced Maryland Automatic Network Disk Archiver (AMANDA), associated with UDP port 10080
Bacula	An open-source network backup server; associated with TCP ports 9101, 9102, and 9103
Bacula Client	Client for the Bacula server; associated with TCP port 9102
DNS	Domain Name Service (DNS) server; associated with port 53, using both TCP and UDP protocols
FTP	File Transfer Protocol (FTP) server, associated with TCP port 21
IMAP over SSL	IMAP over the Secure Sockets Layer (SSL) normally uses TCP port 993
IPsec	Associated with UDP port 500 for the Internet Security Association and Key Management Protocol (ISAKMP), along with the ESP and AH transport-level protocols
Mail (SMTP)	Simple Mail Transport Protocol server, such as sendmail or Postfix, using TCP port 25
Multicast DNS (mDNS)	Associated with UDP port 5353 to support the Linux implementation of zero configuration networking (zeroconf), known as Avahi
NFS4	NFS version 4 uses TCP port 2049, among others
Network Printing Client	The standard print client uses UDP port 631, based on the Internet Print Protocol (IPP)
Network Printing Server	The standard print server client uses TCP and UDP ports 631, based on the Internet Print Protocol (IPP)
OpenVPN	The open-source Virtual Private Network system, which uses UDP port 1194
POP-3 over SSL	POP-3 over the Secure Sockets Layer (SSL) normally uses TCP port 995
RADIUS	The Remote Authentication Dial In User Service (RADIUS) protocol uses UDP ports 1812 and 1813
Red Hat Cluster Suite	The Red Hat suite for multiple systems uses TCP ports 11111 and 21064, along with UDP ports 5404 and 5405
Samba	The Linux protocol for communication on Microsoft networks uses TCP ports 139 and 445, along with UDP ports 137 and 138
Samba Client	The Linux protocol for client communication on Microsoft networks uses UDP ports 137 and 138
Secure WWW (HTTPS)	Communications to a secure web server uses TCP port 443
SSH	The SSH server uses TCP port 22
TFTP	Communications with the Trivial File Transfer Protocol (TFTP) server requires TCP port 69
TFTP Client	Strangely enough, no open port is required for a TFTP client; all communications proceed over the open TCP port 69 through the TFTP server
Virtual Machine Management	Remote access to KVM-based VMs use TCP port 16509
Virtual Machine Management (TLS)	Remote access to KVM-based VMs use TCP port 16509 and can be configured with Transport Layer Security (TLS)
WWW (HTTP)	The well-known web server uses TCP port 80

Grundlegende SELinux Konzepte

SELinux wurde von der U.S. National Security Agency entwickelt um eine rollenbasierte Zugriffskontrolle in Linux einzuführen. SELinux sorgt dafür, dass Sicherheitslücken in einem subsystem wie ftp oder http keinen Einfluss auf den Rest des Systems nehmen kann.

SELinux weist jeder Datei unterschiedliche Kontexte zu, bekannt als subjects, objects und actions. Bei dem Subjekt handelt es sich um einen Prozess wie ein Kommando, eine Aktion oder ein Service wie z.B. der apache web server. Das Objekt ist eine Datei und die Aktion gibt an, was dem Subjekt gegenüber dem Objekt erlaubt ist. So kann dem subject Apache Web Server gegenüber dem object .html-Datei erlaubt werden diese anzuzeigen.

Den SELinux Kontext einer Datei kann man ganz einfach mit ls -Z sehen.

```
# ls -Z
-rw-------. root root system_u:object_r:admin_home_t:s0 anaconda-ks.cfg
-rw-r--r--. root root system_u:object_r:admin_home_t:s0 install.log
-rw-r--r--. root root system_u:object_r:admin_home_t:s0 install.log.syslog
```

In diesem Kapitel beschränken wir uns darauf die für die RHCSA prüfungsrelevanten Teile zu besprechen, als da wären: Ändern des SELinux Modus auf permissive bzw. enforcing; Ausgeben und identifzieren von Datei- und Prozesskontexten; Wiederherstellung des default Dateikontextes und Mit booleschen (wahr/falsch) Schaltern die systemweiten SELinux Einstellungen modifizieren.

SELinux Status

SELinux wird in der /etc/sysconfig/selinux konfiguriert und kann entweder auf enforcing, permissive oder disabled stehen. Im permissive Modus werden Verstöße nicht blockiert wie im enforcing Modus, sondern lediglich mitgeloggt.

Befindet sich SELinux im enforcing Modus hat man die Wahl zwischen zwei verschiedenen Betriebsarten: targeted (einfache Sicherheit und default) und mls (Multi-Level Security). Letzterer erweitert SELinux noch um verschiedene Sicherheitsstufen (siehe /etc/selinux/targeted/setrans.conf)
von c0 bis c3 (top secret) und ist nur in Hochsicherheitsumgebungen sinnvoll, dazu muss das selinux-policy-mls Paket installiert sein.

```
# cat /etc/sysconfig/selinux
# 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 these two values:
#     targeted - Targeted processes are protected,
#     mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

Den aktuellen Status von SELinux kann man mit getenforce oder detailliert mit sestatus sehen:

```
# sestatus
SELinux status:                 enabled
SELinuxfs mount:                /selinux
Current mode:                   enforcing
Mode from config file:          enforcing
Policy version:                 24
Policy from config file:        targeted

# getenforce
Enforcing
```

Mit setenforce kann man selinux on the fly ändern, diese ändern die boolesche /selinux/enforce variable (1 = enforcing, 0 = permissive):

```
# setenforce enforcing
# setenforce permissive
```

Konfiguration regulärer Benutzer in SELinux

Mit dem id -Z Kommando kann man sich seinen eigenen Sicherheitskontext anschauen und wenn man semanage installiert hat (aus dem Paket policycoreutils-python.x86_64) kann man mit semanage login -l alle user Kontexte im Detail sehen:

```
# semanage login -l

Benutzername:             SELinux-Benutzer          MLS/MCS-Bereich

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

# id -Z
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

Was für den root user okay sein mag sollte für reguläre user allerdings geändert werden, da unconfined auf einen undefinierten Zustand hinweist.

Um den user Kontext eines einzelnen users zu ändern kann man das semanage Kommando wieder heranziehen:

```
[root@centosvm ~]# semanage login -a -s user_u mleimenm

[root@centosvm ~]# semanage login -l

Benutzername:             SELinux-Benutzer          MLS/MCS-Bereich

__default__               unconfined_u              s0-s0:c0.c1023
mleimenm                  user_u                    s0
root                      unconfined_u              s0-s0:c0.c1023
system_u                  system_u                  s0-s0:c0.c1023
```

```
[root@centosvm ~]# semanage login -a -s staff_u mleimenm

[root@centosvm ~]# semanage login -l

Benutzername:             SELinux-Benutzer          MLS/MCS-Bereich

__default__               unconfined_u              s0-s0:c0.c1023
mleimenm                  staff_u                    s0-s0:c0.c1023
root                      unconfined_u              s0-s0:c0.c1023
system_u                  system_u                   s0-s0:c0.c1023
```

Die Unterschiede zwischen user_u und staff_u bestehen darin, dass user_u nur Zugriff auf low security (s0) objekte erhält wenn der mls Modus aktiv ist und der staff_u user darüber hinaus Zugriff auf das su und das sudo Kommando erhält und auf alle MLS Stufen Zugriff erhält.

Will man den default Wert für alle zukünftigen User ändern kann man dazu auch wieder auf semanage zurückgreifen:

```
[root@centosvm ~]# semanage login -m -S targeted -s „user_u“ -r s0 __default__
```

In diesem Falle modifiziert (-m) semanage den targeted policy store (-S) mit der SELinux user (-s) user_u mit der MLS s0 range (-r) für den __default__ (vorne und hinten 2 Unterstriche) user. (-:

```
[root@centosvm ~]# semanage login -l

Benutzername:             SELinux-Benutzer          MLS/MCS-Bereich

__default__               user_u                    s0
mleimenm                  staff_u                   s0-s0:c0.c1023
root                      unconfined_u              s0-s0:c0.c1023
system_u                  system_u                  s0-s0:c0.c1023
```

Typische SELinux Benutzerrollen

Benutzerrolle	Beschreibung
guest_u	No GUI, no networking, no access to the su or sudo commands
xguest_u	GUI, networking only via the Firefox web browser
user_u	GUI and networking available
staff_u	GUI, networking, and the sudo command available
unconfined_u	Full system access

Verwalten von booleschen SELinux Einstellungen

Die meisten der SELinux Einstellungen sind boolesche Schalter, die entweder on (1) oder off (0) sein können und befinden sich im /selinux/booleans Verzeichnis, sollten aber über die getsebool und setsebool Kommandos abgefragt und gesetzt werden. Will man die Änderungen auch nach einem reboot behalten wollen sollte man sie persistent (setsebool -P) setzen. Um eine Liste aller bekannten Schalter zu sehen empfiehlt sich ein getsebool -a.

Mit Hilfe dieser Variablen kann man sehr filigrane Einstellungen vornehmen und zum Beispiel den usern Zugriff auf einzelne Kommandos wie z.B. ping (user_ping), ftp oder rsync erlauben oder auch wie im folgenden Beispiel die Ausführungsrechte eigener Skripte im homeverzeichnis des Nutzers oder /tmp verbieten:

```
# setsebool allow_user_exec_content off
```

Um Beschreibungen für die einzelnen Schalter zu bekommen empfielt sich wieder semanage:

```
# semanage boolean -l
```

Anzeigen und identifizieren von SELinux Dateikontexten

Mit dem ls -Z Kommando kann man sich den SELinux Kontext von Dateien ausgeben lassen, schauen wir uns mal ein paar unterschiedliche Kontexte an:

```
[root@centosvm ~]# ls -dZ anaconda-ks.cfg /tmp/ /var/ftp/pub/
-rw-------. root root system_u:object_r:admin_home_t:s0     anaconda-ks.cfg
drwxrwxrwt. root root system_u:object_r:tmp_t:s0           /tmp/
drwxr-xr-x. root root system_u:object_r:public_content_t:s0 /var/ftp/pub/
```

Zuerst sehen wir den normalen ls -l output, dass wir die normalen ugo/rwx Schalter haben, gefolgt von einem . der auf SELinux Kontexte allgemein hinweist (ein + würde auf ACLs hindeuten) und dass die Dateien allesamt dem root user und der root Gruppe gehören.

Darauf folgt der eigentlich interessante Teil was SELinux betrifft, nämlich die „User:Rolle:Typ:MLS-Sicherheitsstufe“ Angaben bzgl. des Kontextes der Dateien.
Der User ist meistens entweder system_u oder unconfined_u und hat üblicherweise keinen Einfluss auf den Dateizugriff. In den meisten Fällen haben Dateien als Rolle object_r gesetzt, auch wenn es sehr wahrscheinlich ist, dass SELinux in Zukunft eine feinere Unterteilung dieser beiden Kategorien trifft. Der eigentliche Schlüsselwert ist aktuell der Typ in den oberen Fällen admin_home_t, tmp_t und public_content_t. Will man das /var/ftp/pub Verzeichnis auch für die Öffentlichkeit beschreibbar machen wollen würde man stattdessen public_content_rw_t setzen.

Um den Kontext einer Datei zu setzen dient das chcon Kommando:

```
# chcon -R -u system_u -t public_content_rw_t /var/ftp/pub
```

Um den Kontext von einer anderen Datei zu übernehmen kann man auch den --reference Parameter nutzen:

```
# chcon -R --reference /var/tmp /tmp
```

Wiederherstellen von SELinux Dateikontexten
Für jede Datei hat SELinux default Kontexten unter /etc/selinux/targeted/contexts/files/file_contexts gespeichert. Möchte man diese defaults wieder herstellen, weil man z.B. Drittherstellern kurzzeitige root Rechte geben musste und sicherstellen will, dass die Dateirechte wieder alle in Ordnung sind, hilft einem das restorecon Kommando (evtl. mit -R für rekursiv und -F für den force-mode) weiter:

```
[root@centosvm ~]# chcon -t public_content_rw_t /var/ftp/pub

[root@centosvm ~]# ls -dZ /var/ftp/pub
drwxr-xr-x. root root system_u:object_r:public_content_rw_t:s0 /var/ftp/pub

[root@centosvm ~]# restorecon /var/ftp/pub

[root@centosvm ~]# ls -dZ /var/ftp/pub
drwxr-xr-x. root root system_u:object_r:public_content_t:s0 /var/ftp/pub
```

Werden neue Dateien oder Verzeichnisse erstellt, so erben diese den Kontext, der in der file_contexts Datei für diese definiert wurde.

Identifizieren von Prozess Kontexten

Genau wie Dateien verfügen auch Prozesse über einen Kontext, den man sich mit ps -Z anzeigen lassen kann:

```
[root@centosvm ~]# ps -eZ
LABEL                              PID TTY          TIME CMD
system_u:system_r:init_t:s0          1 ?        00:00:00 init
system_u:system_r:kernel_t:s0        2 ?        00:00:00 kthreadd
...
system_u:system_r:hald_t:s0       1651 ?        00:00:00 hald-addon-acpi
system_u:system_r:automount_t:s0 1671 ?        00:00:01 automount
system_u:system_r:mcelog_t:s0     1687 ?        00:00:00 mcelog
system_u:system_r:sshd_t:s0-s0:c0.c1023 1699 ? 00:00:00 sshd
system_u:system_r:postfix_master_t:s0 1775 ?   00:00:00 master
system_u:system_r:postfix_qmgr_t:s0 1781 ?     00:00:00 qmgr
system_u:system_r:abrt_t:s0-s0:c0.c1023 1799 ? 00:00:00 abrtd
system_u:system_r:crond_t:s0-s0:c0.c1023 1807 ? 00:00:01 crond
system_u:system_r:crond_t:s0-s0:c0.c1023 1818 ? 00:00:00 atd
system_u:system_r:certmonger_t:s0 1832 ?       00:00:00 certmonger
system_u:system_r:initrc_t:s0     1876 ?       00:00:00 prltoolsd
system_u:system_r:initrc_t:s0     1884 ?       00:00:48 prltoolsd
system_u:system_r:getty_t:s0      1893 tty1    00:00:00 mingetty
system_u:system_r:getty_t:s0      1895 tty2    00:00:00 mingetty
system_u:system_r:udev_t:s0-s0:c0.c1023 2548 ? 00:00:00 udevd
system_u:system_r:dhcpc_t:s0     12852 ?       00:00:00 dhclient
system_u:system_r:sshd_t:s0-s0:c0.c1023 13366 ? 00:00:00 sshd
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 13369 ? 00:00:00 sshd
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 13370 pts/0 00:00:00 bash
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 13394 pts/0 00:00:00 su
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 13400 pts/0 00:00:00 bash
system_u:system_r:postfix_pickup_t:s0 14336 ?  00:00:00 pickup
system_u:system_r:kernel_t:s0   14562 ?        00:00:00 flush-253:1
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 14580 pts/0 00:00:00 ps
```

Wie wir sehen können ändern sich user und role nicht allzu häufig(system_u:system_r für System und unconfined_u:unconfined_r für die persönlichen Prozesse des root users), aber der Typ entspricht häufig dem jeweiligen Dienst.

Diagnostizieren und adressieren von Verstössen gegen die SELinux Richtlinien

Wenn man auf Probleme mit SELinux stösst sollte man es nicht gleich disablen, es gibt im Gegenteil hervorragende Tools um das Problem aufzuspüren, dass sich in den allermeisten Fällen um falsches labeling, Kontextprobleme oder boolesche Schalter handelt.

SELinux Audits

Probleme werden unter /var/log/audit/audit.log getrackt. Da diese Datei gerade für Einsteiger aber recht unübersichtlich ist gibt es verschiedene Tools um diese auszuwerten, z.B. ausearch und sealert.

```
# ausearch -m avc -c su
----
time->Mon Apr 29 14:14:22 2013
type=SYSCALL msg=audit(1367237662.959:27797): arch=c000003e syscall=1 success=no exit=-13 a0=3
a1=7f13bb69ad90 a2=5c a3=0 items=0 ppid=2259 pid=2260 auid=500 uid=0 gid=0 euid=0 suid=0 fsuid=0 egid=0
sgid=0 fsgid=0 tty=pts0 ses=1 comm="su" exe="/bin/su" subj=staff_u:staff_r:staff_sudo_t:s0-s0:c0.c1023
key=(null)
type=AVC msg=audit(1367237662.959:27797): avc:  denied  { compute_av } for  pid=2260 comm="su"
scontext=staff_u:staff_r:staff_sudo_t:s0-s0:c0.c1023 tcontext=system_u:object_r:security_t:s0
tclass=security
```

ausearch filtert das logfile nach bestimmten Kriterien, in diesem Falle nach unerlaubten sudo Zugriffen. Mit -m avc filtert man nach Access Vector Cache Einträgen (-m für messages) und mit -c kann man entsprechende Dienste wie su, httpd, sudo etc. angeben.

Das sealert Tool aus dem setroubleshoot Paket ist da noch etwas aussagekräftiger:

```
# sealert -a /var/log/audit/audit.log
100% donefound 1 alerts in /var/log/audit/audit.log
--------------------------------------------------------------------------------

SELinux is preventing /bin/su from compute_av access on the security .

*****  Plugin catchall (100. confidence) suggests  **************************

If you believe that su should be allowed compute_av access on the  security by default.
Then you should report this as a bug.
You can generate a local policy module to allow this access.
Do allow this access for now by executing:
# grep su /var/log/audit/audit.log | audit2allow -M mypol
# semodule -i mypol.pp
```

SELinux Label- und Kontext-Probleme

SELinux Probleme mit booleschen Schaltern

Das GUI SELinux Management Tool

Um das ***system-config-selinux*** zu nutzen muss das Paket *policycoreutils-gui* installiert sein.
Auch das ***sealert*** tool verfügt mit dem ***sealert -b*** über eine wunderbare GUI-Variante.

Ausgewählte Boolesche SELinux Optionen

Boolescher Schalter	Beschreibung
fcron_crond	Supports fcron rules for job scheduling
cron_can_relabel	Allows cron jobs to change the SELinux file context label
allow_daemons_use_tty	Lets service daemons use terminals as needed
allow_daemons_dump_core	Supports writing of core files to the top-level root directory
init_upstart	Allows supplanting of SysVInit with upstart
allow_mount_anyfile	Permits the use of the mount command on any file
qemu_use_nfs	Supports the use of NFS filesystems for virtual machines
qemu_use_usb	Supports the use of USB devices for virtual machines
qemu_full_network	Supports networking for virtual machines
qemu_use_cifs	Supports the use of CIFS (Common Internet File System) filesystems for virtual machines
qemu_use_comm	Supports a connection for virtual machines to serial and parallel ports
allow_sysadm_exec_content	Allows sysadm_u users the right to execute scripts
allow_xguest_exec_content	Allows xguest_u users the right to execute scripts
allow_user_exec_content	Allows user_u users the right to execute scripts
allow_staff_exec_content	Allows staff_u users the right to execute scripts
allow_guest_exec_content	Allows guest_u users the right to execute scripts

SELinux Szenarien und Lösungen

Lösungsansätze im Problemfall

Problem	Lösungsvorschlag
A file can't be read, written to, or executed.	Review current ownership and permissions with the ls -l command. Apply ownership changes with the chown and chgrp commands. Apply permission changes with the chmod command.
Access to a secure file required for a single user.	Configure ACLs for the appropriate filesystem and then apply the setfacl command to provide access.
The SSH service is not accessible on a server.	Assuming the SSH service is running (a RHCE requirement), make sure the firewall supports SSH access with the iptables -L command; revise as needed with the system- config-firewall tool.
Enforcing mode is not set for SELinux.	Set enforcing mode with the setenforce enforcing command.
Need to restore SELinux default file contexts on a directory.	Apply the restorecon -F command to the target directory.
Unexpected failure when SELinux is set in enforcing mode.	Use the sealert -a /var/log/audit/audit.log command or the SELinux Troubleshooter to find more information about the failure; sometimes a suggested solution is included.
Need to change SELinux options for a user.	Apply the setsebool -P command to the appropriate boolean setting.

Der Boot-Prozess

Dieses Kapitel beschäftigt sich damit, was in dem Zeitraum zwischen dem anschalten des Systems und der Verfügbarkeit des login prompts, eigentlich passiert. Nach der Installation von RHEL6 verweist das BIOS/UEFI auf ein bestimmtes Medium. Nehmen wir einmal an, dass es sich bei dem Medium um die interne Festplatte handelt, so verweist der MBR (Master Boot Record) des Laufwerks wiederum auf den GRUB bootloader. Sobald eine Option um RHEL zu starten im Menü des GRUB ausgewählt wurde startet dieser den Kernel, der anschliessend den init Prozess aufruft. Der init Prozess initalisiert wie der Name schon andeutet das System und versetzt es in einen festgelegten Runlevel, in dem es diverse Services startet wie z.B. den NTP (Network Time Protocol) Client.

Prüfungsrelevante Themen

Verstehen des Bootprozesses

- Boot, reboot, and shut down a system normally
- Boot system into different runlevels manually
- Use single-user mode to gain access to a system
- Configure systems to boot into a specific runlevel automatically
- Configure network services to start automatically at boot
- Modify the system boot loader

Netzwerk Zeit Service (NTP)

- Configure a system to run a default configuration NTP server and synchronize time using other NTP peers.

Das BIOS und UEFI

Auf den meisten modernen Computersystemen hat das UEFI (Unified Extensible Firmware Interface) das bereits deutlich in die Jahre gekommene BIOS (Basic Input/Output System) abgelöst. Auch wenn das UEFI wesentlich leistungsfähiger als das BIOS ist, so gleicht sich ihre Arbeitsweise in Bezug auf den Bootprozess sehr, so dass wir sie für dieses Kapitel synonym nutzen können.

Basiskonfiguration des Systems

Nach dem Einschalten des Systems wird als erstes das BIOS/UEFI gestartet. Grundlegende Einstellungen werden aus dem ROM gelesen und basierend darauf eine Reihe von Hardware Tests gestartet, die man gemeinhin „Power On Self Test (POST)“ nennt. Findet das System Fehler beim POST werden Fehler, sofern die Grafikkarte noch nicht initialisiert wurde, als Piepsen wiedergegeben. Verfügt das System über ein UEFI Menü, so ist es möglich, dass es über ein Trusted-Platform Modul (TPM) verfügt, ist dies der Fall wird es von RHEL6 auch genutzt. Anhand der oben genannten Einstellungen wird auch das Bootdevice bestimmt und die Kontrolle an den Master Boot Record (MBR) bzw. die GUID Partition Table (GPT) dieses Gerätes übergeben. Dort befindet sich neben den Partitioniereungsinformationen üblicherweise die erste Stage des GRUB Bootloaders, die wiederum auf das Bootmenü des GRUB verweist, welches dem Anwender nun angezeigt wird, bevor nach Ablauf eines Timeouts ein default Eintrag gestartet wird.

Auf älteren BIOS Systemen gibt es noch die Einschränkung, dass der bootloader (und damit die /boot Partition) in den ersten 1024 Zylindern des Laufwerks befinden muss. Ausserdem muss sich das Bootlaufwerk bei mehren Laufwerken entweder am primären Controller (im Falle von PATA) bzw. auf einem Laufwerk mit der ID 0 oder 1 (SCSI) befinden muss.

Bootloader und GRUB

Der Standard bootloader unter RHEL 6 ist GRUB 0.97, der sich grundlegend in der Konfiguration vom GRUB2 (>2.0) unterscheidet, auch wenn man aus Anwendersicht keinen grossen Unterschied sehen mag. Möchte man die Sicherheitsfunktionen von TPM nutzen empfiehlt sich die Installation des TrustedGRUB als Ersatz für den regulären.

GRUB kann man nicht nur starr durch die Konfigurationsdateien bearbeiten, es ist sogar möglich aus dem Bootmenü selbst heraus noch temporäre Änderungen für den aktuellen Bootvorgang vorzunehmen oder sogar eine eigene GRUB shell aufzurufen, von der aus man noch zusätzliche Informationen abrufen und komplette Einträge neu erstellen kann.

Booten in verschiedene Runlevel

Wenn man im GRUB Menü die Taste „a“ für append drückt kann man dem Kernel weitere Parameter mitgeben (für eine Liste lohnt der Besuch von https://www.kernel.org/doc/Documentation/kernel-parameters.txt). Die Zeile mit dem Kernelaufruf könnte ungefähr so aussehen:

```
grub append> ro root=UUID=somelonghexadecimalnumber rd_NO_LUKS rd_NO_LVM rd_NO_MD rd_NO_DM LANG=en_US.UTF-8
SYSFONT=latarcyrheb-sun16 KEYBOARDTYPE=pc KEYTABLE=us crashkernel=auto rhgb quiet
```

Um in verschiedene Runleven zu booten fügt man einfach die Nummer des Runlevels (1-5) bzw. für den single user mode das Stichwort „single“ an.

Runlevel Kernel Parameter unter RedHat

Runlevel Boot Parameter	Beschreibung
single	Single user mode that does NOT execute any startup scripts from /etc/rc1.d
init=/bin/sh	ignores init setup completely and invokes only a shell instead of any startup scripts
0	power off (do not set this as the boot runlevel)
1	Single User mode which executes all startup scripts in the /etc/rc1.d directory
2	*user defineable custom runlevel*
3	Textbased multiuser mode
4	*user defineable custom runlevel*
5	GUI based multiuser mode
6	reboot (do not set this as the boot runlevel)

Modifizieren des GRUB bootloaders

Um den Prüfungspunkt „modify the system bootloader“ erfüllen zu können muss man den Aufbau der GRUB Konfigurationsdatei im Detail kennen. Eine typische /boot/grub/grub.conf Datei sieht zum Beispiel folgendermassen aus:

```
# grub.conf generated by anaconda
#
# Note that you do not have to rerun grub after making changes to this file
#
# NOTICE:  You have a /boot partition.  This means that
#          all kernel and initrd paths are relative to /boot/, eg.
#          root (hd0,0)
#          kernel /vmlinuz-version ro root=/dev/mapper/vg0-rootvol
#          initrd /initrd-[generic-]version.img
# boot=/dev/sda

default=0
timeout=5
splashimage=(hd0,0)/grub/splash.xpm.gz
hiddenmenu

title CentOS (2.6.32-358.2.1.el6.x86_64)
        root (hd0,0)
        kernel /vmlinuz-2.6.32-358.2.1.el6.x86_64 ro root=/dev/mapper/vg0-rootvol rd_NO_LUKS rd_LVM_LV=vg0/
swapvol  KEYBOARDTYPE=pc KEYTABLE=de-latin1-nodeadkeys rd_NO_MD rd_LVM_LV=vg0/rootvol SYSFONT=latarcyrheb-
sun16 crashkernel=auto LANG=de_DE.UTF-8 rd_NO_DM rhgb quiet
        initrd /initramfs-2.6.32-358.2.1.el6.x86_64.img

title CentOS (2.6.32-358.el6.x86_64)
        root (hd0,0)
        kernel /vmlinuz-2.6.32-358.el6.x86_64 ro root=/dev/mapper/vg0-rootvol rd_NO_LUKS rd_LVM_LV=vg0/
swapvol  KEYBOARDTYPE=pc KEYTABLE=de-latin1-nodeadkeys rd_NO_MD rd_LVM_LV=vg0/rootvol SYSFONT=latarcyrheb-
sun16 crashkernel=auto LANG=de_DE.UTF-8 rd_NO_DM rhgb quiet
        initrd /initramfs-2.6.32-358.el6.x86_64.img
```

Gehen wir die Konfigurationsdatei mal Schritt für Schritt durch. Ganz oben im Kommentarbereich erfahren wir, dass die Datei bei der Installation von Anaconda erstellt wurde und dass wir wenn wir Änderungen in der Datei vorgenommen haben nicht zwingend das grub Kommando ausführen müssen um den MBR neu zu schreiben. Früher bei den sogenannten single-stage Bootloadern war dies zwingend erforderlich da die komplette Konfiguration in die ersten 512 Zylinder des Bootmediums geschrieben werden musste, da beim GRUB dank der multi-stage Architektur in der ersten Stage lediglich der Verweis auf die eigentliche Konfiguration (stage 2) im /boot Filesystem liegt müssen Änderungen also nicht in den MBR kopiert werden.

Danach folgt, sofern wir über eine getrennte /boot Partition verfügen, der Hinweis, dass alle Pfadangaben relativ vom /boot Verzeichnis und nicht vom / selbst liegen. Dies führt manchmal zu Verwirrungen, aber dazu gleich mehr.
Anschliessend folgt ein generisches Beispiel für einen einzelnen Eintrag im Bootmenü und in der letzten Kommentarzeile ein Hinweis auf welchem Bootmedium der MBR liegt, bzw. zum Zeitpunkt der Installation lag (im Beispiel /dev/sda).

Doch nun zu den eigentlichen Einstellungen:

```
default=0
```

Wie man an den darunterfolgenden Abschnitten sehen kann gliedert sich die Konfiguration der einzelnen Menüeinträge in Abschnitte, die aus mehreren Zeilen bestehen und mit dem Stichwort „title“ beginnen. Diese werden von 0 aus numeriert, so dass 0 im oberen Beispiel auf CentOS (2.6.32-358.2.1.el6.x86_64) und 1 auf CentOS (2.6.32-358.el6.x86_64) verweist. Mit dem Stichwort default legt man fest welcher Eintrag nach Ablauf des Timeouts automatisch gebootet wird.

```
timeout=5
```
Hier wird ein timeout von 5 Sekunden gesetzt. Trifft der Anwender in dieser Zeit keine Auswahl im Menü wird der default Eintrag gebootet.

```
splashimage=(hd0,0)/grub/splash.xpm.gz
```
Unter dem splashimage versteckt sich der Dateiname eines gezippten Bildes, welches im Hintergrund des Bootmenüs angezeigt werden kann.

```
hiddenmenu
```
Ist der Schalter hiddenmenu gesetzt, so wird das Bootmenü nicht automatisch dargestellt, sondern gibt dem User einen kurzen Zeitraum in dem er falls er es denn möchte das Menü aufrufen kann, ansonsten wird einfach der default Eintrag gebootet.

Im Anschluss folgen wie bereits erwähnt die eigentlichen Einträge. Für Linux Systeme bestehen diese Abschnitte üblicherweise aus den folgenden Einträgen:

```
title CentOS (2.6.32-358.2.1.el6.x86_64)
```
Die title Direktive legt fest unter welchem Namen dieser Eintrag im Menü erscheinen soll.

```
        root (hd0,0)
```
Jetzt wird es wie versprochen ein wenig verwirrend. Auch wenn diese Direktive root heisst, so verweist sie doch stattdessen auf die /boot Partition (siehe Kommentarbereich), die in diesem Beispiel auf der ersten Partition 0 der ersten Festplatte hd0 liegt. Genausogut könnte die /boot Partition auch in der UUID Notation angegeben werden. Die UUID Notation ist die für RHEL6 empfohlene Methode, sowohl für die grub.conf als auch für die fstab. Der Eintrag sieht dann in etwa so aus:

```
        root=UUID=16stellige_Hexadezimalzahl
```
UUID ist ein Akronym für den Universally Unique Identifier, eine 128bit grosser Wert, der in Hexadezimal angegeben wird und eindeutig für jedes Volume im System generiert wird. Die LABEL Direktive aus RHEL 5 sollte nicht mehr genutzt werden.

```
        kernel /vmlinuz-2.6.32-358.2.1.el6.x86_64 ro root=/dev/mapper/vg0-rootvol rd_NO_LUKS rd_LVM_LV=vg0/
swapvol  KEYBOARDTYPE=pc KEYTABLE=de-latin1-nodeadkeys rd_NO_MD rd_LVM_LV=vg0/rootvol SYSFONT=latarcyrheb-
sun16 crashkernel=auto LANG=de_DE.UTF-8 rd_NO_DM rhgb quiet
```
Mit der kernel Direktive wird der zu bootende Kernel sowie alle an diesen zu übergebenden Parameter festgelegt. Die Position ist wieder abhängig von der oben genannten root Direktive, sprich in inserem Falle liegt der Kernel unter /boot/vmlinuz*. ro legt fest, dass der Kernel nur readonly genutzt um versehentliche Schreibzugriffe aus der initial ramdisk zu unterbinden. Anschliessend folgt der root Kernel Parameter, der nicht mit der GRUB Direktive root verwechselt werden sollte, sondern auf das tatsächliche /-Filesystem verweist. Mit den rd_ Parametern werden bestimmte Feature explizit initialisiert oder unterdrückt. rd_NO_LUKS sorgt dafür, dass keine mit dem Linux Unified Key Setup (LUKS) verschlüsselten Dateisysteme unterstützt werden. rd_LVM_LV definiert zum booten wichtige Logical Volumes, während ein rd_NO_LVM im Gegenzug die Unterstützung von Logical Volumes zum booten vollständig unterbunden hätten. rd_NO_DM und rd_NO_MD streicht die Unterstützung von RAID-Volumes. Mit KEYBOARDTYPE, LANG, SYSFONT und KEYTABLE legt man das default Environment in Bezug auf Sprache, Tastatureinstellung und Schriftart fest. Der crashkernel Parameter legt die Grösse des für den Crashkernel reservierten Speichers fest und sollte auf auto stehen, lediglich einige wenige alte RHEL6 Systeme benötigen einen fixen Wert stattdessen, so dass man dort z.B. crashkernel=128M setzt. Zu guter letzt unterdrücken die Paramter rhgb und quiet die Ausgabe der Bootmeldungen.

```
        initrd /initramfs-2.6.32-358.2.1.el6.x86_64.img
```
Die initrd Direktive legt den Ort der initialen Ramdisk Images fest. Auch hier ist die Angabe relativ zur root Direktive zu sehen, sprich das Image liegt hier unter /boot/initramfs-*.img.
Die initiale RAM-Disk erzeugt beim booten ein temporäres Dateisystem, welches Kernel Module und Userspace Programme enthält, die zum mounten der restlichen Dateisysteme und zum Starten des Systems erforderlich sind.

Befindet sich z.B. auf einem Notebook eine Dual-Boot Installation zusammen mit Windows so finden sich meist noch die folgenden Zeilen in der grub Konfiguration:

```
title Windows 7
        rootnoverify (hd0,1)
        chainloader +1
```

Hier wird ein Menüeintrag mit dem Namen „Windows 7“ hinzugefügt, welches auf der ersten Festplatte in der 2. Partition liegt. Im Gegensatz zu dem root Eintrag wird mit rootnoverify vom Grub nicht wie bei Linux üblich überprüft. Mit Hilfe von chainloader +1 wird die Kontrolle an den ersten Sektor der Partition übergeben von wo aus Windows den Boot Prozess weiterführt.

Weitere hilfreiche Optionen kann man der *kernel-parameters.txt* Datei im *kernel-doc* Paket entnehmen. Gängig sind z.B. mem=xxxM, welches in dem Falle, dass das System nicht die korrekte Menge an RAM erkennt weiterhilft, oder vga=791 um das Display auf 1024x768 x 16bit festzusetzen, wenn es Probleme mit der Grafikkarte gibt.

Muss der MBR neu geschrieben werden so kann man dazu einfach ***grub-install*** ausführen, bei Änderungen in der Konfigurationsdatei sind keine weiteren Schritte nötig um diese zu „aktivieren“.

GRUB Sicherheit und Passwortschutz

Wenn ein RedHat System in den Single-User Modus bootet wird er Benutzer nicht nach dem Passwort gefragt und wird direkt als root User angemeldet. Dies stellt natürlich ein ernstzunehmendes Sicherheitsrisiko dar. Um das System gegen unberechtigten Zugriff zu schützen hat man daher noch weitere Möglichkeiten, zum einen kann man das BIOS/UEFI mit einem Passwort schützen, so dass der Anwender die Bootgeräte und -reihenfolge nicht ändern kann und zum anderen kann man auch Passwörter für den Grub setzen. Hat man bei der Installation von RHEL6 bereits ein Passwort für den Grub mit angegeben, so findet man einen Parameter in der folgenden Art in der Konfiguration des Grub:

```
password --md5 $1$hfBhb8zA$sYrw4B1VzrrpPHpDtyhb.
```

Diese Direktive legt einen Passworthash fest, der mit dem md5 (Message-Digest 5) Verfahren verschlüsselt ist. Um ein neues Passwort festzulegen führt man am besten das grub-md5-crypt Kommando aus, welches eine Passworteingabe erfragt und den entsprechenden hash ausgibt, so dass man diesen per copy&paste in die Konfiguration übertragen kann.

Die Position der password Direktive ist ebenfalls entscheidend. Befindet sie sich im Kopfteil der Konfiguration, also vor den eigentlichen Menüeinträgen, so schützt das Passwort das komplette Grub Menü und unterbindet Änderungen der vorgegebenen Einträge. Befindet sich die password Direktive aber innerhalb eines Menüeintrages, so wird das Passwort benötigt wenn man den Menüpunkt auswählen möchte.

Die Grub Shell

Ein Fehler in der Grub Konfiguration kann dazu führen, dass ein System nicht mehr startet. So kann z.B. ein falscher *root* Eintrag zu einem Kernel-Panic führen. Führt die Grub Konfiguration vor dem Start des Betriebssystems zu Problemen ist meist einer der folgenden Grub Fehler dafür zuständig und man landet in der grub shell:

Grub Fehlermeldungen

Meldung	Beschreibung
Error 15: File not found	Die Partition wurde gemountet, es wurde aber kein Kernel auf der Partition gefunden. Ursache ist wahrscheinlich, dass der root(hdX,partY) Eintrag nicht auf eine gültige /boot Partition verweist.
Error 17: Cannot mount selected partition	Die angegebene Partition enthält kein erkennbares Dateisystem, z.B. wenn root(hdX, partY) auf die swap Partition verweist.
Error 22: No such partition	Die mit root(hdX, partY) angegebene Partition existiert nicht.

Anhang A: Checkliste RHCSA Prüfungsthemen

Understand and use essential Tools

Certification Objective	Study Guide Coverage	Chapter
Access a shell prompt and issue commands with correct syntax		2
Use *grep* and regular expressions to analyze text streams and file		2
Use input/output redirection		2
Access remote systems using SSH		1
Access remote systems using VNC		8
Log in and switch users in multi-user runlevels		7
Archive, compress, unpack, and uncompress files using *tar, star, gzip,* and *bzip2*		8
Create and edit text files		2
Create, delete, copy and move files and directories		2
Create hard and soft links		2
List, set, and change ugo/rwx permissions		3
Locate, read, and use system documentation including man, info, and files in */usr/share/doc*		2

Operate Running Systems

Certification Objective	Study Guide Coverage	Chapter
Boot, reboot, and shut down a system normally		4
Boot systems into different runlevels manually		4
Use single user mode to gain access to a system		4
Identify CPU/Memory intensive processes, adjust process priority with *renice*, and *kill* processes		8
Locate and interpret system log files		8
Access a virtual machine's console		1
Start and stop virtual machines		1
Start, stop, and check the status of network services		2

Create and Configure Filesystems

Certification Objective	Study Guide Coverage	Chapter
Create, mount, unmount, and use ext2, ext3, and ext4 file systems		5
Mount, unmount, and use LUKS-encrypted filesystems		5
Mount and unmount CIFS and NFS network filesystems		5
Configure systems to mount ext4, LUKS-encrypted and network filesystems automatically		5
Extend existing unencrypted ext4-formatted logical volumes		5
Configure and set-GID directories for collaboration		7
Create and manage Access Control Lists (ACLs)		3
Diagnose and correct file permission problems		3

Deploy, Configure and Maintain Systems

Certification Objective	Study Guide Coverage	Chapter
Configure networking and hostname resolution statically or dynamically		2
Schedule tasks using *cron*		8
Configure systems to boot into a specific runlevel automatically		4
Install Red Hat Enterprise Linux automatically using Kickstart		1
Configure a physical machine to host virtual guests		1
Install Red Hat Enterprise Linux systems as virtual guests		1
Configure systems to launch virtual machines at boot		1
Configure a system to run a default configuration NTP server and synchronize time using other NTP peers		RHCE-7
Configure network services to start automatically at boot		4
Configure a system to run a default configuration HTTP server		E
Configure a system to run a default configuration FTP server		E
Install and update software packages from Red Hat Network, a remote repository, or from the local filesystem		6
Update the kernel package appropriately to ensure a bootable system		6
Modify the system bootloader		4

Manage Users and Groups

Certification Objective	Study Guide Coverage	Chapter
Create, delete, and modify local user accounts		7
Change passwords and adjust password aging for local user accounts		7
Create, delete, and modify local groups and group memberships		7
Configure a system to use an existing LDAP directory service for user and group information		7

Manage Security

Certification Objective	Study Guide Coverage	Chapter
Configure firewall settings using *system-config-firewall* or *iptables*		3
Set enforcing and permissive modes for SELinux		3
List and identify SELinux file and process context		3
Restore default file contexts		3
Use boolean settings to modify system SELinux settings		3
Diagnose and address routine SELinux policy violations		3