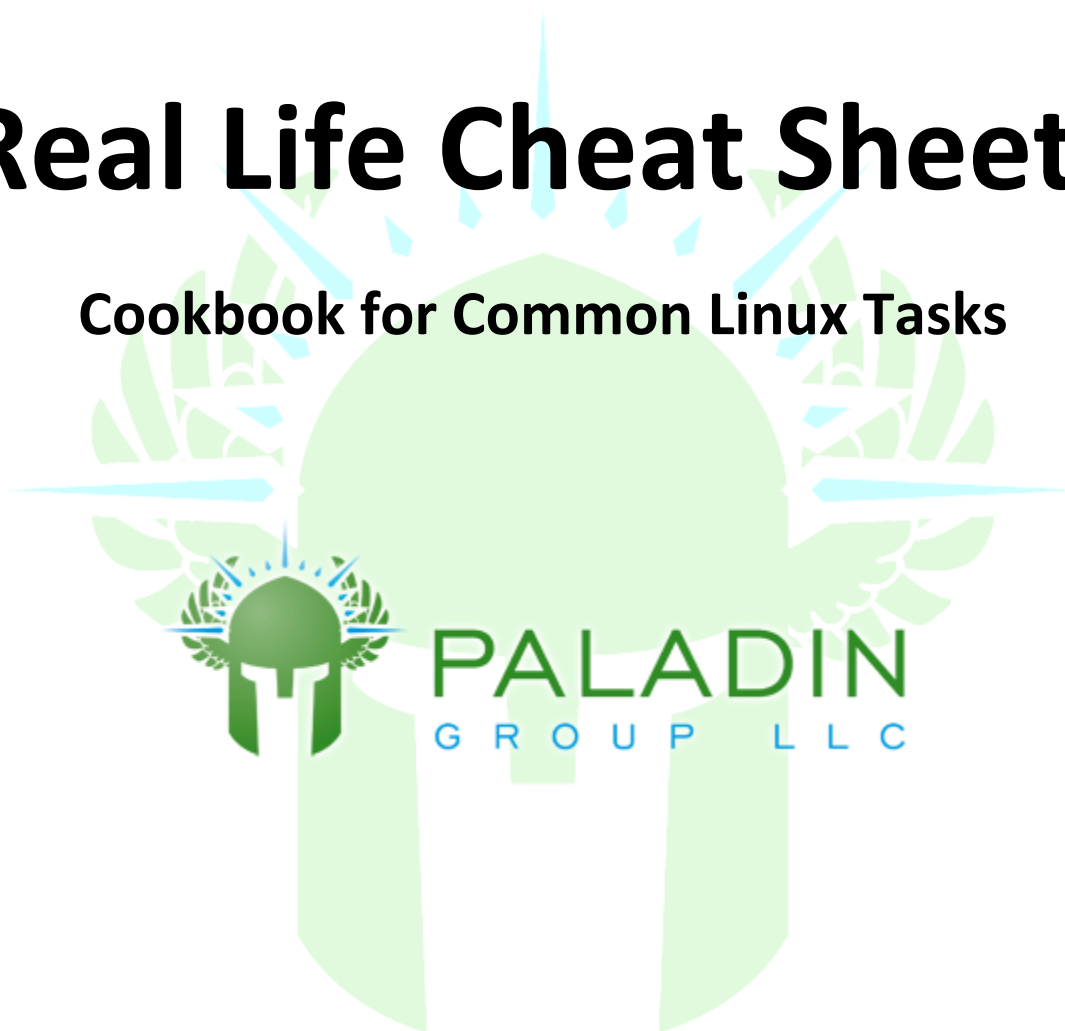


# Real Life Cheat Sheets

Cookbook for Common Linux Tasks





## REAL LIFE CHEAT SHEET – Setting a GRUB password

### GRUB passwords

You can easily bypass all system security by simply booting to single user mode, if you have physical access to the workstation or server. The **GRUB** boot loader allows you to set a password so that users cannot alter the boot command line without the password.

The command to create a grub password is

#### **grub-md5-crypt**

```
[root@workstation1 ~]# grub-md5-crypt
Password:
Retype password:
$1$2c4o/$gMHLZBZZqJT8oRPpiShkf/
```

Once you have the md5 hash of the password you need to copy it and insert it into the **grub.conf** file

The format of the line is below (put this before the "default=" line)

```
password --md5 results_of_grub_md_5_command
```

#### **/etc/grub.conf after a password has been added**

```
[root@workstation1 ~]# cat /etc/grub.conf
# 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/sda2
#   initrd /initrd-version.img
#boot=/dev/sda
password --md5 $1$2c4o/$gMHLZBZZqJT8oRPpiShkf/
default=0
timeout=5
splashimage=(hd0,0)/grub/splash.xpm.gz
hiddenmenu
title CentOS (2.6.18-194.el5)
    root (hd0,0)
    kernel /vmlinuz-2.6.18-194.el5 ro root=LABEL=/ rhgb quiet
    initrd /initrd-2.6.18-194.el5.img
```



## REAL LIFE CHEAT SHEET – Getting into a system without the root or GRUB password

1. Insert your Linux Install disk in the CDROM drive
2. Enter the computer's BIOS, then configure the computer to boot to the CDROM drive first
3. When you get the "boot:" prompt, type **linux rescue**
4. When asked "What language would you like to use during the installation process?", choose **English**
5. When asked "What type of keyboard do you have" choose **us**
6. When asked if you want to start networking interfaces on this system, choose **No**
7. When asked "The rescue environment will now attempt to fine your Linux installation and mount it under the directory /mnt/sysimage..." choose **Continue**
8. When prompted that "Your system has been mounted under /mnt/sysimage" choose **OK**
9. You should now be at a shell prompt#, type **chroot /mnt/sysimage**
10. Reset the password with the command

**passwd**

the system will prompt you to choose a new password and type it twice

11. now use vi or nano to edit the file **/boot/grub/grub.conf**
  - a. remove the line starting **password --md5 ....**
  - b. save the file
12. reboot your computer, notice there is no longer a GRUB password and you have reset the root password and can log in again.



## REAL LIFE CHEAT SHEET – Repairing a broken MBR

1. Insert your Linux Install disk in the CDROM drive (always keep a install/rescue disk for each version of linux handy, like next to your linux computer)
2. Enter the computer's BIOS, then configure the computer to boot to the CDROM drive first
3. When you get the "boot:" prompt, type **linux rescue**
4. When asked "What language would you like to use during the installation process?, choose **English**
5. When asked "What type of keyboard do you have" choose **us**
6. When asked if you want to start networking interfaces on this system, choose **No**
7. When asked "The rescue environment will now attempt to fine your Linux installation and mount it under the directory /mnt/sysimage..." choose **Continue**
8. When prompted that "Your system has been mounted under /mnt/sysimage" choose **OK**
9. You should now be at a shell prompt#, type **chroot /mnt/sysimage**
10. Type **grub**  
This will put you in interactive **grub** mode. Follow the rest of the session below (the commands you type are highlighted AND bold)

```
GNU GRUB version 0.97 (640K lower / 3072K upper memory)
[ Minimal BASH-like line editing is supported. For the first word, TAB
  lists possible command completions. Anywhere else TAB lists the possible
  completions of a device/filename.]

grub> root (hd0,0)
File system type is ext2fs, partition type 0x83

grub> setup (hd0)
Checking if "/boot/grub/stage1" exists... no
Checking if "/grub/stage1" exists... yes
Checking if "/grub/stage2" exists... yes
Checking if "/grub/e2fs_stage1_5" exists... yes
Running "embed /grub/e2fs_stage1_5 (hd0)"... 15 sectors are embedded.
succeeded
Running "install /grub/stage1 (hd0) (hd0)1+15 p (hd0,0)/grub/stage2 /grub/grub
.conf"... succeeded
Done.

grub> quit
```

11. type **exit** to exit your chroot shell
12. eject your cdrom
13. type **exit** again to reboot
14. verify your system boots

**IMPORTANT** this assumes your “/boot” partition is the first partition on the first hard drive (/dev/sda1) which is almost always is. If for some reasons it's NOT you have to modify your **root(hd0,0)** line



## REAL LIFE CHEAT SHEET – Creating a partition and formatting it with an ext3 file system

To create a partition and format it with the **ext3** file system

1. **fdisk /dev/disk\_specifier**                      ex. `fdisk /dev/sda`
2. type **p** to print out the partition table, note the LAST partition identifier
3. type **n** to create a NEW partition
4. when prompted for a start cylinder, just hit the **return** key
5. when prompted for the end cylinder, type  
**+100M**                      (to create a 100M partition, a 1G partition would be +1G)
6. type **p** to print the new partition table, note the **/dev/xxx** identifier of the new partition... you need this later. write it here \_\_\_\_\_
7. type **w** to write your change and exit fdisk
8. type **partprobe** to inform the kernel of the new partitions
9. create a new file system with the command  
**mkfs -t ext3 /dev/xxx** (where **/dev/xxx** is the partition you created and listed in step 6)
10. create a "mount point" where you want the directory to be grafted onto the file system tree  
**mkdir /mnt/mynewpartition**
11. mount the file system  
**mount /dev/xxx /mnt/mynewpartition**
12. verify it's mounted with  
**df -h**

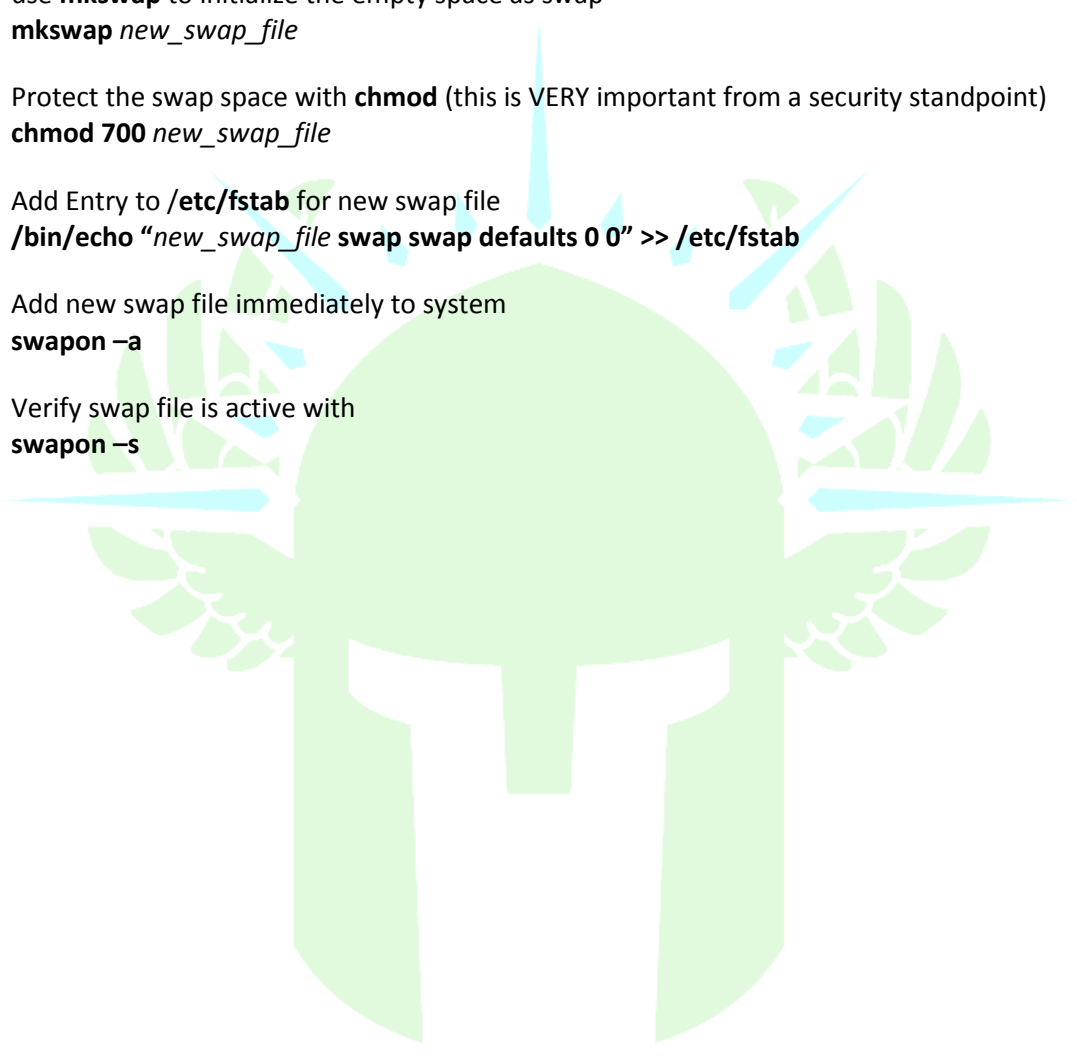
Remember to have it automatically mount on reboots, you need to edit `/etc/fstab` and add a new line to reference the new partitions. If for example if our new partition is `/dev/sda7` and we want to mount it as `/mnt/mynewpartition` we could type the following command to add it to the system

```
[root@workstation1 ~]# echo "/dev/sda7 /mnt/mynewpartition ext3 defaults 1 2" >> /etc/fstab
```



## REAL LIFE CHEAT SHEET – Adding a swap file

### Creating a swap file

1. use **dd** to create a empty space  
**dd if=/dev/zero of=new\_swap\_file ibs=1M count=size\_in\_megabytes**
  2. use **mkswap** to initialize the empty space as swap  
**mkswap new\_swap\_file**
  3. Protect the swap space with **chmod** (this is VERY important from a security standpoint)  
**chmod 700 new\_swap\_file**
  4. Add Entry to **/etc/fstab** for new swap file  
**/bin/echo "new\_swap\_file swap swap defaults 0 0" >> /etc/fstab**
  5. Add new swap file immediately to system  
**swapon -a**
  6. Verify swap file is active with  
**swapon -s**
- 



## REAL LIFE CHEAT SHEET – Creating a LUKS encrypted file system

1. Create a partition as normal, for the purpose of this cheat sheet, we will call it /dev/sda
2. Determine a “/dev/mapper name”. For the purpose of this class we will call it encrypted\_data
3. Create a mount point, for the purpose of this cheat sheet we will call it /usr/mysecretdata  
**mkdir /usr/mysecretdata**
4. Create random data on the partition you just created (this is optional and it can take a LONG LONG LONG time. However it is very good from a security standpoint)  
**dd if=/dev/urandom of=/dev/sda7**
5. Initialize the partition for encryption  
**cryptsetup luksFormat /dev/sda7**
6. Tell the encryption software to start using encryption on the partition, and create a special encrypted block device on the underlying physical partition.  
**cryptsetup luksOpen /dev/sda7 encrypted\_data**
7. Create a Linux usable file system on the special encrypted block device  
**mkfs -t ext3 /dev/mapper/encrypted\_data**
8. Add the following line to **/etc/fstab** so your disk will automatically mount at system startup.  
**/dev/mapper/encrypted\_data /usr/mysecretdata ext3 defaults 1 2**
9. Inform the system to create the special encrypted block device on system startup.  
Edit (create if necessary) **/etc/crypttab**, add the following line  
**encrypted\_data /dev/sda7 none**
10. Mount the disk, so you can immediately use it without rebooting (or reboot if you prefer)  
**mount /dev/mapper/encrypted\_data /usr/mysecretdata**  
or  
**reboot**



## REAL LIFE CHEAT SHEET – Enabling Quotas

1. Edit **/etc/fstab** to mount the file system with the options **usrquota, grpquota** or both.

```
[root@workstation1#] nano /etc/fstab
----- /etc/fstab (before) -----
LABEL=/1      /          ext3 defaults 1 1

-----/etc/fstab (after) -----
LABEL=/1      /          ext3 usrquota,grpquota 1 1
```

2. Remount your file system to take advantage of the new options with the command **mount -o remount file\_system**

```
[root@workstation1#] mount -o remount /
```

3. Build the quota database with the command: **quotacheck -cugm file\_system**

```
[root@workstation1#] quotacheck -cugm /
```

4. Turn on quotas with the command **quotaon file\_system**

```
[root@workstation1#] quotaon /
```

5. Set your EDITOR variable to use your favorite editor

```
[root@workstation1#] export EDITOR=nano
```

6. Assign disk quotas with the command: **edquota username\_to\_assign\_quotas\_to**

```
[root@workstation1#] edquota user1
```

7. Verify the new quotas have been enabled with the command: **quota -v username\_to\_assign\_quotas\_to**

```
[root@workstation1#] quota -v user1
```

```
Session Edit View Bookmarks Settings Help
Disk quotas for user user60 (uid 500):
Filesystem      blocks      soft      hard      inodes      soft      hard
/dev/sda5       4776         0         0         274         0         0
```

(note in the above screen shoot your file system may differ, and your username would be user1 if you were setting a quota on **user1**)



## REAL LIFE CHEAT SHEET – Connecting to NFS resources

1. Make sure **netfs** is chkconfig'ed on (this is needed to automatically mount NFS shares on reboot)  
**chkconfig --list netfs**

```
[root@workstation1 ~]# chkconfig --list netfs
netfs      0:off 1:off 2:off 3:on 4:on 5:on 6:off
```

2. Use **showmount** to determine what resources are on a server  
**showmount -e ip\_or\_hostname\_of\_remote\_server**

```
[root@workstation1 ~]# showmount -e 192.168.2.186
Export list for 192.168.2.186:
/shared *
/nfshome *
```

3. Make a directory to use as the "graft point"

**mkdir /mnt/a**

4. Mount one of the remote file systems

**mount remote\_server:remote\_path mount\_point**

```
[root@workstation1 ~]# mount 192.168.2.186:/shared /mnt/a
```

5. Verify the remote file system is present on your system now.

**df -h**

```
[root@workstation1 ~]# df -h
Filesystem                Size      Used   Avail   Use%   Mounted on
/dev/sda2                  12G       3.3G    7.6G    30%    /
/dev/sda1                   99M       12M     83M    13%    /boot
tmpfs                      252M        0     252M    0%    /dev/shm
192.168.2.186:/shared    18G       3.4G    14G    21%    /mnt/a
```

6. Add a line to **/etc/fstab** so it always mounts on system boot.

```
[root@workstation1 ~]# echo "192.168.2.186:/shared /mnt/a nfs defaults 0 0" >> /etc/fstab
```



## REAL LIFE CHEAT SHEET – Joining an NIS (YP) domain

NIS is a system that allows you to share “user accounts” and other information across a network. NIS was highly used in Unix/Linux installation for the centralized user management features.

To join an **NIS** domain you must first have the following information

- NIS domain name
- Server IP (optional)

The Steps are

1. make sure **ypbind** is set to run in run level 3 and 5  
**chkconfig --level 35 ypbind on**
2. run **system-config-authentication**
3. click on **enable NIS Support**
4. click on **configure NIS**
5. enter your **NIS Domain Name**
6. enter your **Servers IP address**
7. click **ok**
8. click **ok**
9. close out of **system-config-authentication**

You should now be able to access network accounts via NIS.

One useful command to verify that NIS is working is

**ypcat passwd** which reads the password file from NIS

You also need to make sure your users home directories are available on the system... this is usually done with NFS.



## Real Life Cheat Sheet – Creating a YUM repository (using HTTP)

Creating a YUM repo (http)

- 1) Setup a web server

```
yum install httpd  
service httpd start  
chkconfig --level 35 httpd on
```

```
[root@devel1 ~]# yum install httpd  
Loaded plugins: fastestmirror  
Loading mirror speeds from cached hostfile  
* base: mirror.anl.gov  
Upgrade 0 Package(s)  
... output deleted ...  
Total download size: 1.2 M  
Is this ok [y/N]: y  
Downloading Packages:  
httpd-2.2.3-45.el5.centos.i386.rpm | 1.2 MB 00:01  
Running rpm_check_debug  
Running Transaction Test  
Finished Transaction Test  
Transaction Test Succeeded  
Running Transaction  
Installing : httpd 1/1  
  
Installed:  
httpd.i386 0:2.2.3-45.el5.centos  
  
Complete!  
[root@devel1 ~]# service httpd start  
Starting httpd: httpd: Could not reliably determine the server's fully qualified domain name,  
using 127.0.0.1 for ServerName  
[ OK ]  
[root@devel1 ~]# chkconfig --level 35 httpd on
```

- 2) Install the **createrepo** package

**yum install createrepo**

```
[root@devel1 html]# yum install createrepo
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
* base: mirror.anl.gov
... output deleted ...
Total download size: 59 k
Is this ok [y/N]: y
Downloading Packages:
createrepo-0.4.11-3.el5.noarch.rpm      | 59 kB  00:00
Running rpm_check_debug
Running Transaction Test
Finished Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing      : createrepo              1/1

Installed:
createrepo.noarch 0:0.4.11-3.el5

Complete!
```

- 3) Copy the rpm files to **/var/www/html** (from the CentOS 5.6 CDROM in this case, make sure the CDROM is inserted into the computer or the VMware instance)

**mkdir /var/www/html/myrepo**

**cp -rp /media/CentOS\_5.6\_Final/CentOS/\* /var/www/html/myrepo**

```
[root@devel1 html]# mkdir /var/www/html/myrepo
[root@devel1 html]# cp -rp /media/CentOS_5.6_Final/CentOS/* /var/www/html/myrepo/
```

- 4) Create the rpm listing

**createrepo /var/www/html/myrepo**

```
[root@devel1 html]# createrepo /var/www/html/myrepo
2683/2683 - gthumb-2.7.8-8.el5.i386.rpm
```

- Optional step: create group lists (list of related packages to install at one time). To do this you need to create an xml file to describe which file is in which packages. In this case we'll copy the group list that's provided on the CentOS 5.6 install.

```
cd /var/www/html/myrepo
cp /media/CentOS_5.6_Final/repo/compdata/comps.xml .
createrepo -g comps.xml .
```

```
[root@devel1 html]# cd /var/www/html/myrepo
[root@devel1 /]# cp /media/CentOS_5.6_Final/repo/compdata/comps.xml .
[root@devel1 myrepo]# createrepo -g comps.xml .
2683/2683 - gthumb-2.7.8-8.el5.i386.rpm
Saving Primary metadata
Saving file lists metadata
Saving other metadata
```

- Now you can use your repo. On your clients you need to create a **repo** file in **/etc/yum.repos.d**. Using your favorite editor create a file called **/etc/yum.repos.d/myrepo.repo**

```
[myrepo]
name=CentOS-$releasever - Base
baseurl=http://your_servers_IP/myrepo
gpgcheck=1
enabled=1
```

- Clear your yum cache  
**yum clean all**

```
[root@devel1 yum.repos.d]# yum clean all
Loaded plugins: fastestmirror
Cleaning up Everything
Cleaning up list of fastest mirrors
```

- 8) View your new yum repo  
**yum repolist**

```
[root@devel1 yum.repos.d]# yum repolist
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
repo id          repo name          status
myrepo           CentOS-5 - Base    enabled: 2,683
repolist: 2,683
```

- 9) View the items in your yum repo  
**yum list | grep myrepo**

```
[root@devel1 yum.repos.d]# yum list | grep myrepo
Cluster_Administration-as-IN.noarch 5.2-1.el5.centos myrepo
Cluster_Administration-bn-IN.noarch 5.2-1.el5.centos myrepo
Cluster_Administration-de-DE.noarch 5.2-1.el5.centos myrepo
Cluster_Administration-en-US.noarch 5.2-1.el5.centos myrepo
Cluster_Administration-es-ES.noarch 5.2-1.el5.centos myrepo
Cluster_Administration-fr-FR.noarch 5.2-1.el5.centos myrepo
Cluster_Administration-gu-IN.noarch 5.2-1.el5.centos myrepo
...
zsh.i386 4.2.6-5.el5 myrepo
zsh-html.i386 4.2.6-5.el5 myrepo
```