Tips and tricks

How to become root

Many commands can only be run as the root user, for these commands you will have to become root. This can be accomplished by `su` or by `sudo`

Using su

```
su - <username>

su <username>
```

Both commands will change to the user *username*, however adding "-" to `su` will not only change to the user, but will also change the user's environment, this is the PATH variables and change to the home directory. If no username is specified, then the root user is assumed.

Using sudo

If you prefer not to login as root every time, you want to do any administration, then you need to use `sudo`

`sudo` lets you run a certain set of command as root.

```
toshaan ALL=(ALL) ALL
```

This will allow toshaan to execute all commands as root.

```
sudo <command to run as root>
```

When `sudo` is run, it will ask for a password, this is the password of toshaan. If you do not want all commands, you can specify certain commands or a group of commands.

```
Cmnd_Alias  LOCATE  =  /usr/bin/  ↲
   updatedb
toshaan ALL = LOCATE
```

In this example the user *toshaan* will be able to execute only *updatedb*.

```
Cmnd_Alias NETWORKING =
/sbin/route,
/sbin/ifconfig,
/bin/ping,
/sbin/dhclient,
/usr/bin/net,
/sbin/iptables,
/usr/bin/rfcomm,
/usr/bin/wvdial,
/sbin/iwconfig,
/sbin/mii-tool
toshaan ALL = NOPASSWD: NETWORKING
```

The difference between the previous settings and the current ones is the addition of *NOPASSWD,* thus allowing the user *toshaan* to execute all commands under the *NETWORKING* alias.

```
toshaan    ALL    =    LOCATE,    ↲
   NETWORKING
```

This line would allow both the *NETWORKING* commands and the *LOCATE* commands to be executed.

Securing OpenSSH

To secure OpenSSH more than the default values, the following needs to be changed in */etc/ssh/sshd_config*

- Disable root login:
  `PermitRootLogin no`
- Enable only SSH Protocol 2
  `Protocol 2`
- Change the port to a non-standard port (do not forget to change it in the iptables settings)
  `Port 2345`
- Limit User Login
  `Allowuser <username 1> <username 2>`
- Key based authentication
  `PubkeyAuthentication yes`

```
AuthorizedKeysFile   .ssh/
authorized_keys
```

- On the client create the public and private keys

```
ssh-keygen -t rsa
```

- Setting the permissions correctly because of the *StrictModes* being enabled in the configuration file.

```
chmod 700 ~/.ssh
chmod 600 ~/.ssh/id_rsa
```

- Copy the public key to the remote server and get the contents into the *authorized_keys* file

```
cat id_rsa.pub >> ~/.ssh/
authorized_keys
```

- If you really want to disable password authentication

```
PasswordAuthentication no
```

- Filtering at firewall level with iptables

```
iptables -A INPUT -p tcp -s
81.81.81.81 --dport 22 -j
ACCEPT
```

- Set iptables to only accept traffic from IP address 81.81.81.81 on port 22.

```
iptables -A INPUT -p tcp
--dport 2222 --syn -m limit
--limit 1/m --limit-burst 3
-j ACCEPT
iptables -A INPUT -p tcp
--dport 2222 --syn -j DROP
```

Here these lines limits the number of connections per port per minute to 3, any more connections get dropped.

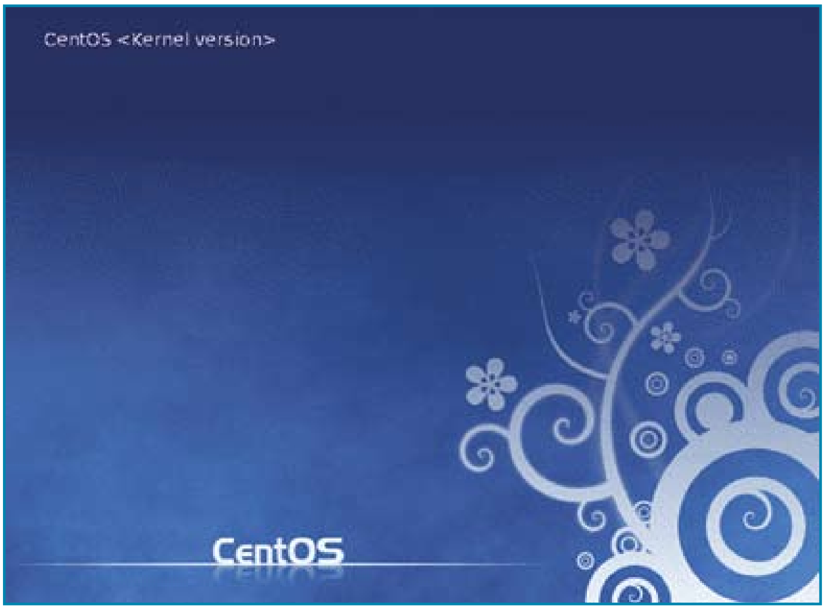
CentOS <Kernel version>

Figure 1. *GRUB boot screen, with the right kernel version and location*

Yum with fastestmirror

Fastestmirror is a plugin for yum. It keeps track of which of the mirrors in the list of mirror list is the fastest and sorts the rest of the list from fast to slow by checking the latency of each mirror. It is highly recommended to enable the fastestmirror plugin, unless you are behind a proxy server.

Installation of the fastmirror plugin is done with the following command: `yum install yum-fastest-mirror`

Now, that the package has been installed, it still needs to be configured. The file that needs to be edited is: */etc/yum/pluginconf.d/fastestmirror.conf*:

```
[main]
verbose = 0
socket_timeout = 3
enabled = 1
hostfilepath=/var/cache/yum/↲
    timedhosts.txt
maxhostfileage = 1
exclude=.gov,   facebook,   ↲
    myspace, junk-mirror.com
```

How to re-install grub boot sector

If your GRUB boot sector get corrupted, for instance when installing other operating systems, you need to reinstall the boot code GRUB.

- Boot from CentOS installation disc. Type “*linux rescue*” at the “*boot:*” prompt.
- Change root to your hard disk root location:

```
chroot /mnt/sysimage
```

- Re-install the GRUB to the hard disk:

```
grub-install /dev/sda
```

- Initialize GRUB by entering into

```
t
```

How to obtain an accurate server time?

The graphical interface is not suited for advanced settings. You are required to setup a small script to synchronise the time:

```
/sbin/service ntpd stop ; / ↲
    usr/sbin/ntpdate -s tick. ↲
    usno.navy.mil &&
/sbin/hwclock -w ; /sbin/ ↲
    service ntpd start
```

Customize your bash prompt

This following customizations give your personal bash prompt a facelift to your settings. All configurations need to be edited in *~/.bashrc*:

```
PS1='\u@\H:\w\$ '
```

This will display the prompt as follows *user@hostname.domain.tld:/working/directory*

For a colourful prompt, you will need to edit the same file *~/.bashrc*:

```
PS1='\[\033[1;32m\]          ↲
    [\[\033[1;34m\]\         ↲
    u@\h\[\033[00m\]\        ↲
    [\033[1;32m\]]\          ↲
    [\033[01;35m\]\W\        ↲
    [\033[1;33m\]$\[\033[00m\]
```

This will display the prompt *[user@hostname]/path/to/directory* in blue with yellow markings and the path in purple. For a full color description, please take a look at: *http://tldp.org/HOWTO/Bash-Prompt-HOWTO/x329.html*

A system-wide configuration is available at: */etc/bashrc*