

## BackTrack Quick Reference Guide

**BackTrack** is the result of the merging of two Innovative Penetration Testing live Linux distributions **Whax** and **Auditor**. Combining the best features from both distributions, and paying special attention to small details, this is probably the best version of either distributions to ever come out. Based on SLAX (Slackware), BackTrack provides user modularity. This means the distribution can be easily customised by the user to include personal scripts, additional tools, customised kernels, etc.

### Step 1:

Boot with backtrack live distro and log on with username **root**, and password **toor**.

### Step 2:

Use **fdisk** to deal with partitions present on the Hard Disk.

### Step 3:

Assuming **hda1** for Linux and **hda2** for swap, do:

```
root@slax:~# fdisk /dev/hda
root@slax:~# mke2fs /dev/hda1
root@slax:~# mkswap /dev/hda2
root@slax:~# swapon /dev/hda2
root@slax:~# mkdir /dev/hda1
root@slax:~# mount /dev/hda1 /mnt/hda1
root@slax:~# cp --preserve -R \
/{bin,dev,etc,home,lib,root,sbin,usr,var,opt,pentest} /mnt/hda1

root@slax:~# mkdir /mnt/hda1/{boot,mnt,proc,sys,tmp}
root@slax:~# cp /boot/boot/vmlinuz /mnt/hda1/boot
root@slax:~# mount --bind /dev /mnt/hda1/dev
root@slax:~# mount -t proc proc /mnt/hda1/proc
root@slax:~# chroot /mnt/hda1 /bin/bash
root@slax:~# liloconfig
```

(follow the GUI - choose simple, your resolution and install to mbr. Lilo will complain and fail but don't worry!)

```
root@slax:~# vi /etc/lilo.conf
```

```
.....
label = back|track
root = current
read-write
.....
```

Save, exit and then

```
root@slax:~# lilo
```

it should then say, added back|track (also Windows if you are dual booting), then unmount and reboot.

Now you are in and backtrack is installed. Let's play with something.

### Change Hostname:

```
root@slax:~# vi /etc/rc.d/rc.M
```

### Disable CDROM eject, cause might be boring ☺ edit the shutdown script:

```
mascalzone ~ # vi /etc/rc.d/rc.6
```

### Fix the so called sexy frame buffer:

Install the following patch: <http://www.remote-exploit.org/splash-fix.mo>, then after booting from hard disk, issue the following commands (bolded):

```
mascalzone ~ # mo2dir /splash-fix.mo /
mascalzone ~ # splash -s -f /etc/bootsplash/themes/Linux/config/bootsplash-
1024x768.cfg > /boot/splash.initrd
mascalzone ~ # vi /etc/lilo.conf
```

Edit your *lilo.conf*, make sure to add the *vga=0x317*, and *initrd* lines. For example:

```
boot = /dev/hda1
prompt
timeout = 20
bitmap=/boot/splash.bmp
change-rules
reset
vga = 0x317
image = /boot/vmlinuz
  root = current
  initrd=/boot/splash.initrd
  label = slax
  read-write
```

Once edited, run *lilo*:

```
mascalzone ~ # lilo -v
mascalzone ~ # reboot
```

### Change the logon screen:

```
mascalzone ~ # vi /etc/issue
```

### Start the lovely tool for bash easy access:

```
mascalzone ~ # yakuake
```

press F12 and go!

### Searching for versions:

```
mascalzone ~ # cat /etc/slax-version
```

```
SLAX 5.0.7
mascalzone ~ # cat /etc/slackware-version
Slackware 10.2.0
```

### Installing application:

```
mascalzone ~ # slapt-get --search gnupg
gnupg-1.2.7-i486-1 [inst=no]: gnupg (The GNU Privacy Guard)
gnupg-1.4.2.2-i486-1 [inst=no]: gnupg (The GNU Privacy Guard)

mascalzone ~ # slapt-get --install gnupg-1.4.2.2-i486-1/mnt/hda1
```

### Installing Slackware packages:

You can install Slackware packages using the command:

```
mascalzone ~ # installpkg package_name.tgz
```

### Converting RPMs to tgz packages:

You can convert *RPM* packages used by distributions like Fedora/Redhat, SuSE, Mandriva, etc to Slackware packages using the command:

```
mascalzone ~ # rpm2tgz package_name.rpm
```

### Startup script:

When BT boots up, */etc/rc.M* gets executed, and runs the other *rc* files in a specific order. *rc.inet1* is used to configure the interfaces and *rc.inet2* is used to start various network services. *rc.local* gets executed towards the end of the process and should be used for initialization of things that need to be started towards the end of the init process.

### To autostart some\_program:

```
mascalzone ~ # cd /root/.kde/Autostart
```

```
mascalzone ~ # ln -s /path_where_program_is/program_name program_name
```

## Networking with BT

If you have a DHCP server, you need to start the DHCP client:

```
mascalzone ~ # dhcpcd
```

this will start DHCP discovery on all cards that support it. To run dhcp on only a specified interface, for instance the first ethernet card (eth0), type:

```
mascalzone ~ # dhcpcd eth0
```

script to configure network, open a terminal and type:

```
mascalzone ~ # netconfig
```

take a look at the routing table with:

```
mascalzone ~ # netstat -nr Or with: mascalzone ~ # route -C -n
```

configure manually network interface with:

```
mascalzone ~ # ifconfig eth0 ip_address_here netmask subnet_mask_here up
```

set the default gateway with:

```
mascalzone ~ # route add default gw ip_address_here
```

set the dns server with:

```
mascalzone ~ # echo "nameserver dns_ip_address_here" > /etc/resolv.conf
```

check interfaces configuration with:

```
mascalzone ~ # ifconfig -a or for wireless: mascalzone ~ # iwconfig
```

to copy files with Windows, temporarily share the Windows hard drive, and then mount it in BT typing:

```
mascalzone ~ # mount -t smbfs -o \  
username=win_suser,password=win_admin_passwd \  
//Win_IP/shared_folder mnt_point
```

## Allowing ICMP (for ping):

```
mascalzone ~ # echo 0 > /proc/sys/net/ipv4/icmp_echo_ignore_all  
mascalzone ~ # echo 0 > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts
```

(this setting can be done also with `sysctl -w net.ipv4.icmp_echo_ignore_all=0`)

(-----< END >-----)