

The Open Pitt



What's cooking in Linux and Open Source in Western Pennsylvania

Issue 24

May 2006

www.wplug.org

Hacking Knoppix, Part 2 by Sean Parsons

Sean finishes his adventures with the book *Knoppix Hacks: 100 Industrial-Strength Tips and Tools* by Kyle Rankin, published by O'Reilly Media <<http://www.oreilly.com/catalog/knoppixhks/>>.

Last month, we learned what Knoppix is, some tricks to make it a more useful desktop system, and how to install it to your hard drive. Now we'll look at some of the more advanced ways it can be used.

Hack #49: Wipe a Hard Drive

Frequently, second-hand hard drives bought using eBay or at swap meets contain easily-recoverable sensitive files. Simply deleting the information isn't enough to keep it from prying eyes. Knoppix comes with a tool called *shred* to make it more difficult for others to recover data from an old hard drive that you are getting rid of. To securely wipe your hard drive, boot into Knoppix and find the partition on your desktop that you want to shred. Then open a terminal and shred it with this command:

```
$ sudo shred -n 2 -z -v /dev/hda1
```

Replace */dev/hda1* with the partition(s) holding the data you want to obliterate.

What I used Hack #49 for: A friend of mine just purchased a new computer and wanted to donate her old PC to a charity. But she didn't want to donate all her personal data, too. I just gave her a copy of Knoppix and a photocopy of this hack.

Hack #53: Repair the GRUB Bootloader

There are two popular bootloaders for Linux: LILO and GRUB. GRUB has many interesting features including the ability to change kernels at boot time. Sometimes, GRUB gets overwritten when installing Microsoft Windows or by an accidental installation of LILO to the hard drive's master boot record (MBR). The procedure to restore GRUB is fairly simple.

First, identify your Linux root partition. In this example the root partition is */dev/hda1*. Next, while running from your Knoppix live CD, you'll need to mount the partition with the *dev* option enabled and with write permission. If the filesystem is not yet mounted, mount it using the following command in a terminal.

```
$ sudo mount -o dev,rw /mnt/hda1
```

If the filesystem is already mounted, you can remount it with:

```
$ sudo mount -o remount,dev,rw /mnt/hda1
```

Once this is done, restore GRUB with the following command:

```
$ sudo chroot /mnt/hda1 grub-install /dev/hda
```

What I used Hack #53 for: One of my brothers recently managed to overwrite his MBR when reinstalling Windows. Thanks to Knoppix, it was easy to reinstall GRUB and restore his ability to dual-boot.

Hack #94: Create a Customized Knoppix

One reason there are so many Knoppix-based distributions is that they are so easy to create. With this hack, you will soon be running your own custom version of Knoppix.

First, boot from your Knoppix CD. Most of the commands you use to re-master the CD require root privileges; instead of typing *sudo* in front of everything, click the menu item *KNOPPIX » Root Shell* to launch a terminal with root privileges.

Next, you'll need to have a disk space to work in. Set up a partition formatted with a Linux filesystem. To use the *bootfrom* cheat code (from Hack #5 last month), make sure you use a filesystem compatible with *bootfrom*—I recommend *ext2*. Plan to have at least 4.5 GB of free space (you can get away with only 3 GB of space if you have 1 GB of RAM).

See *KNOPPIX*, p. 2

April Roundup

Apr. 1 General User Meeting: **Ted Rodgers** discussed running Linux on the PowerPC architecture. Although now being abandoned by Apple, it is still extensively used in IBM servers and special-purpose embedded devices. He covered the challenges of booting using Open Firmware and how to set up and partition a hard drive. Ted also answered questions on accessing PCI and USB devices, enabling 3D video acceleration, and available applications.

Apr. 29 Special Presentation: Virtualization—which lets a single computer appear to be several machines, each running an independent operating system—is today's hot topic. **Bob Good** of **VMware** spoke about the uses of his company's virtualization products for consolidating servers, ensuring high availability, and disaster recovery. This includes the ability to migrate a running virtual machine from one machine to another, as well as central management of all machines.

Coming Events

May 17 (Wednesday): General User Meeting, Topic: Cryptography. 6:30PM to 9PM, 5409 Wean Hall, CMU

May 27: Installfest. 10AM to 5PM, Newell-Simon Hall 3rd floor atrium (Perlis Atrium), CMU

The public is welcome at all events

Listing 1: Determining your system's available memory

```

root@ttypl[knoppix]# free
              total        used         free   shared  buffers   cached
Mem:           321784      317040       4744         0       3812    112888
-/+ buffers/cache:  200340      121444
Swap:          433712       12476      421236

```

KNOPPIX, from p. 1

Now we should check available RAM, as we need a minimum of 1 GB of physical memory plus swap space. Listing 1 is what my system shows when I run *free* in a terminal.

The "total" column only adds up to about 737 MB, so we need to increase this by creating additional swap space. I chose to do this by making a swap file to give me an extra 750 MB.

```

# dd if=/dev/zero
of=swapfile bs=1M count=750
750+0 records in
750+0 records out
786432000 bytes
transferred in 27.858599
seconds (28229417 bytes/sec)
# mkswap swapfile
setting up swapspace
version 1, size = 786427 kB
# swapon swapfile

```

At this point, you'll need to copy the complete Knoppix filesystem to your hard drive. Change to the target directory on your hard drive and use the commands:

```

# mkdir source
# mkdir source/KNOPPIX
# cp -Rp /KNOPPIX/*
source/KNOPPIX

```

This will take a while (approximately 10 minutes). Next you'll need to copy your *resolv.conf* file over so that your new Knoppix image can connect to the Internet.

```

# cp -f /etc/dhccp/resolv.conf
source/KNOPPIX/etc/dhccp/resolv.conf

```

Now *chroot* into the Knoppix directory on the hard drive and mount the *proc* filesystem.

```

# chroot source/KNOPPIX
# mount -t proc /proc proc

```

At this point you can start using *apt-get* to add and remove various packages. You can also modify your sources located at */etc/apt/sources.list* to be able to install packages that are not currently in your repositories. Once you are ready to create a CD based on your changes, clear out the cache of packages you have downloaded to conserve space:

```

# apt-get clean

```

If you intend for your system to fit on a standard CD, try to make sure your customized filesystem is no bigger than the original. Before exiting the *chroot* environment, make sure you unmount the *proc* filesystem with:

```

# umount /proc

```

Then press Ctrl-D to exit the environment. Here is a quick rundown of the commands I used to create my master CD filesystem.

```

# mkdir master
# rsync -a --exclude
"/KNOPPIX/KNOPPIX" /cdrom/
master/
# mkisofs -R -U -V "Knoppix
Hacks filesystem" -P "Knoppix
Hacks" -hide-rr-moved -cache-inodes
-no-bak -pad source/KNOPPIX | nice
-5 /usr/bin/create_compressed_fs
- 65536 > master/KNOPPIX/KNOPPIX

```

By far this is the most time consuming command, not just because it will take you a while to type it in, but because it is compressing your filesystem. It took me close to 45 minutes to compress it, and you can safely ignore the warnings it outputs about creating a filesystem that does not conform to ISO-9660.

Now that you've customized your master directory, you need to update your MD5 checksums prior to creating your CD image:

```

# cd master
# rm -f KNOPPIX/md5sums
# find -type f -not -name
md5sums -not -name boot.cat
-exec md5sums {} \; >>
KNOPPIX/md5sums
# cd ..

```

Now to create the CD image:

```

# mkisofs -pad -l -r -J -v
-V "KNOPPIX" -noemul-boot
-boot-load-size 4
-boot-info-table -b
boot/isolinux/isolinux.bin
-c boot/isolinux/boot.cat
-hide-rr-moved -o
knoppix.iso master/

```

You now have a new *knoppix.iso* image that you can burn to CD.

What I Used Hack #94 For: I really found myself addicted to this hack. I

The Open Pitt is published by
the Western Pennsylvania Linux
Users Group
<<http://www.wplug.org/top/>>

Editors:

Elwin Green
Vance Kochenderfer

What is Linux?

Linux is a *kernel*, the core of a computer operating system, created by Linus Torvalds. It is typically packaged as a *distribution*, which includes the extra programs necessary to make a computer functional and useful. Since 1991, it has grown from a one-man project which ran on one computer to one with thousands of contributors running on everything from personal organizers to million-dollar supercomputers.

What are Open Source and Free Software?

Open Source and Free Software provide you, the user, with the opportunity to see the source code of the programs you use. You are free to use it, share it with others, and even make changes to it if you wish. While the Free Software and Open Source communities differ in their philosophical approach, in practical terms they share nearly identical goals. Learn more at <<http://www.opensource.org/>> and <<http://www.gnu.org/>>.

This newsletter was produced using Open Source and Free Software.

Copyright 2006 Western Pennsylvania Linux Users Group. Any article in this newsletter may be reprinted elsewhere in any medium, provided it is not changed and attribution is given to the author and WPLUG.

kept reusing this and then moved onto remastering Morphix, a Knoppix derivative. By the time I was done, I had an installable live CD based on Ubuntu with a GNOME desktop and the ability to use Java, Flash, and listen to MP3 files. I even managed to include my own personalized desktop theme that I called p9l.

Conclusion

I've barely touched on the many well thought out and clearly explained hacks included in this book. This is probably one of the best technical books I've ever read and so I would strongly recommend it to others as well.

Sean Parsons has been happily hacking on GNU/Linux for over four years and is very excited to give back to the community through WPLUG.