CPU
COMPUTER POWER USER

Tips & Tricks

Click To Print

Tips & Tricks
January 2008 • Vol.8 Issue 1
Page(s) 98-99 in print issue

Warm Up To Penguins

The Down & Dirty Details Of Linux Network Configuration, Part III

For the last two months, we've been looking at the networking tricks you can do from the command line. In our last installment, it’s time to see how your network is configured when you boot your system, talk about DHCP, and discuss wireless networking.

Obviously, you don’t need to manually configure your network when Linux boots. It happens as a natural part of the boot process. Here are all the details. There may be minor differences on other distros, but you’ll be getting the “truth according to Ubuntu.”

The first thing you need to be familiar with is what happens when Linux boots. Without getting into all the messy details about Grub, LILO, and run levels, the simple version of the story is that at some point in the boot, a script in /etc/init.d is run. This script, appropriately enough, is called “networking.” There’s not much in there, apart from the normal scaffolding needed by all init.d scripts. (They need to be able to respond to start, stop, and restart requests, which requires some shell scripting to handle.)

In fact, all the script really does is run the ifup command. In this case, ifup means “bring up an interface.” The version in the networking script says “ifup –a,” which means “bring up all interfaces.” But what do we mean by “all”?

The answer to that question lies a few directory levels over, in /etc/network. There is a file there labeled “interfaces,” and it has the configuration information for all the network interfaces that should automatically come up during boot. Here’s a sample output:

```
auto loiface lo inet loopback

iface eth0 inet staticaddress 192.168.1.220netmask 255.255.255.0gateway 192.168.1.1auto eth0
```

What does this mean? The first two lines tell ifup to automatically bring up the loopback interface (the one located at 127.0.0.1). The next four lines tell the system to create a static IP address on the eth0 interface, at the given local address, netmask, and gateway through the given IP address. The last line instructs the script to bring up the eth0 interface automatically (when the –a option is used with ifup). Without the auto command, you’ll need to bring up the interface manually.

■ The Dynamic Duo

The other method of configuring an interface, DHCP, has to keep a process running because a DHCP lease needs to be renewed periodically. To let the networking script start up the DHCP process, you need to change the network configuration file. If you have a system with an /etc/network/interfaces file, you’d change the entry to “iface eth0 inet dhcp.” Alternatively, if your network configuration is in /etc/sysconfig/network-scripts, you’d add a line that read “BOOTPROTO=dhcp.”

If you have an interface that you want to configure with DHCP for one-time use (for example, if you have a laptop and you’re using it at a company you’re visiting), you can start DHCP from the command line. If you are trying to run DHCP on the device eth0, you merely have to type (as root, or using sudo) **dhclient eth0**.

You can use dhclient without any arguments; in this case, it looks at the file /etc/dhclient.conf for advice. This file can be very simple (or even empty), or it can be quite complex, in the case of some of the more elaborate DHCP configurations of some corporations. For any system that will be regularly using DHCP, just modify the network script and let the network startup take care of the details.

■ The How & Why Of Wi-Fi

Historically, Wi-Fi has been one of the hardest things to configure with Linux. In the beginning, many Linux distributions wouldn't support some network cards out of the box, and you had to go through a complicated and perilous kernel patch and recompile to get them to work.

Recently, support for the most common Wi-Fi cards and chipsets are built into the kernel. You still need to configure it to get a successful connection, however. And if there's one place we recommend using the GUI configuration, it would be for Wi-Fi. Distributions such as Ubuntu ship with slick Wi-Fi tools that rival those found on Windows. They will scan for available hotspots, handle WEP and WPA keys, and generally keep the details out of your hair.

If you plan to use your laptop at a hotspot or wireless router that you're unfamiliar with, play it safe and use the GUI networking tools.

If you can't get to the GUI, you can still configure your wireless connection, but it's going to be painful. To begin, you'll want to find out what networks are available. You'll want to use the iwlist command. If your wireless card is on ath0, you can get a list of all the access points in the area by typing (as root) **iwlist ath0 scan**. The output will look something like this:

```
ath0 Scan completed : Cell 01 - Address: 00:12:17:46:E6:AF ESSID:"cpuap" Protocol:IEEE
802.11bg Mode:Master Channel:1 Encryption key:off Bit Rate:1 Mb/s Bit Rate:2 Mb/s Bit Rate:5.5
Mb/s Bit Rate:6 Mb/s Bit Rate:9 Mb/s Bit Rate:11 Mb/s Bit Rate:12 Mb/s Bit Rate:18 Mb/s Bit
Rate:24 Mb/s Bit Rate:36 Mb/s Bit Rate:48 Mb/s Bit Rate:54 Mb/s Quality=82/100 Signal level=-
48 dBm Extra: Last beacon: 36ms ago
```

In this case, we can see a single access point in range, with the ESSID of "cpuap," available on channel 1 with no encryption and a reasonably good signal strength.

With this information in hand, you use the iwconfig command (think ifconfig, but with a "W" for wireless) to actually set up the connection. For the access point shown in the previous example, type the following in the command line: **iwconfig ath0 essid cpuap channel 1**.

This will make the ath0 interface available for connection, but you'll still need to use ifconfig to set up the actual network details (and possibly dhclient for DHCP support). If your network uses WEP encryption, you would need to add more information, making the command "iwconfig ath0 essid cpuap channel 1 key 23F0-332A-2341-563B," assuming that 23F0-332A-2341-563B was the WEP password in hex format. You can't use WEP passphrases with iwconfig.

Suppose that you are using WPA encryption instead of WEP. At this point, unless you have a lot of time on your hands and want to become an expert of wireless security protocols, you *really* want to use the GUI tools. Getting WPA to work from the command line involves invoking a program called wpa_supplicant. This isn't for the faint of heart; you can set up WPA through the command line, but it can be a time-consuming process, even for seasoned vets.

■ A Farewell To Networking

This ends our three-month odyssey to the mysterious land of Linux command line networking. You may not be ready to become a Linux network administrator, but if your GUI ever goes south on you, you may have some hope of getting your network working again. And even if you never leave the comfort of a GUI, it never hurts to know what's happening behind the scenes.

by James Turner

Infinite Loop: The Real Da Vinci Code

Granted, this latest discovery isn't quite of the same magnitude as unearthing a deep-rooted, far-reaching conspiracy, but it might be more believable. Italian musician Giovanni Maria Pala believes that he has uncovered a hidden soundtrack in Leonardo da Vinci's "The Last Supper." Further study of the painting also reveals a secret message—"Michelangelo stinks"—as well as da Vinci's prediction for Super Bowl XLV.

www.sunherald.com/152/story/197036.html