Wireless networking

iwconfig

Wireless networking is somewhat more complicated than traditional wired networking, and requires additional tools for setup. Slackware includes a diverse collection of wireless networking tools to allow you to configure your wireless network interface card (WNIC) at the most basic level. We won't cover everything here, but should give you a solid foundation to get up and running quickly. The first tool we are going to look at is ***iwconfig***(8). When run without any argument, ***iwconfig*** displays the current wireless information on any and all NICs on your computer.

```
darkstar:~# iwconfig
lo        no wireless extensions.

eth0      no wireless extensions.

wmaster0  no wireless extensions.

wlan0     IEEE 802.11abgn  ESSID:"nest"
          Mode:Managed  Frequency:2.432 GHz  Access Point:
00:13:10:EA:4E:BD
          Bit Rate=54 Mb/s   Tx-Power=17 dBm
          Retry min limit:7   RTS thr:off   Fragment thr=2352 B
          Encryption key:off
          Power Management:off
          Link Quality=100/100  Signal level:-42 dBm
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:0  Invalid misc:0   Missed beacon:0

tun0      no wireless extensions.
```

Unlike wired networks, wireless networks are *"fuzzy"*. Their borders are hard to define, and multiple networks may overlap one another. In order to avoid confusion, each wireless network has (hopefully) unique identifiers. The two most basic identifiers are the Extended Service Set Identifier (ESSID) and the channel or frequency for radio transmission. The ESSID is simply a name that identifies the wireless network in question; you may have heard it referred to as the *"network name"* or something similar.

Typical wireless networks operate on 11 different frequencies. In order to connect to even the most basic wireless network, you will have to setup these two pieces of information, and possibly others, before setting up things like the WNIC's IP address. Here you can see that my ESSID is set to *"nest"* and my laptop is transmitting at 2.432 GHz. This is all that is required to connect to an unencrypted wireless LAN. (For any of you out there expecting to come to my house and use my unencrypted wireless, you should know that you'll have to break a 2048-bit SSL key before the access point will let you communicate with my LAN.)

```
  darkstar:~# iwconfig wlan0 essid nest \
```

```
  freq 2.432G
```

The *freq* and *channel* arguments control basically the same thing. You only need to use one. If you are unsure what frequency or channel to use, Slackware can usually figure this out for you.

```
  darkstar:~# iwconfig wlan0 essid nest \
  channel auto
```

Now Slackware will attempt to connect to the strongest access point on the *"nest"* essid operating at any frequency.

Wired Equivalent Protection (or Lack Thereof)

Wireless networking is by its very nature less secure than wired networking. Having your information travelling on the airwaves makes it highly susceptible to interception by third parties, so over the years a number of methods have been devised to make wireless networking more secure. The first was called Wired Equivilant Protection, or WEP for short, and fell far short of its goal. If you are still using WEP today, I encourage you to consider using WPA2 or some other form of stronger encryption. Attacks against WEP are trivial and take only minutes to perform. Unfortunately there are still access points configured for WEP, and you may need to connect to one from time to time. Connecting to WEP encrypted access points is fairly simple, particularly if you have the key in hexidecimal format. We'll need to pass the *key* argument along with the password in hexidecimal or ASCII format. If using an ASCII password, you'll need to prepend it with *"s:"* but generally speaking, hexidecimal format is preferred.

```
darkstar:~# iwconfig wlan0 \
  key cf80baf8bf01a160de540bfb1c
darkstar:~# iwconfig wlan0 \
  key s:thisisapassword
```

Wifi Protected Access

Wifi Protected Access (or WPA for short) was the successor for WEP that aimed to fix several problems with wireless encryption. Unfortunately, WPA had some flaws as well. An update called WPA2 offers even stronger protection. At this time, WPA2 is supported by nearly all wireless network cards and access points, but some older devices may only support WEP. If you need to secure your wireless network traffic, WPA2 should be considered the minimum level of protection required. Unfortunately, ***iwconfig*** is unable to setup WPA2 encryption on its own. For that, we need a helper daemon, ***wpa_supplicant***(8).

Unfortunately, there's no easy way to manually configure a WPA2 protected network; you'll have to edit `/etc/wpa_supplicant.conf` directly with a text editor. Here we will discuss the simplest form of WPA2 protection, the Pre-Shared Key, or PSK for short. For details on setting up Slackware to connect to more complicated WPA2 encrypted networks, see the man page for `wpa_supplicant.conf`.

```
# /etc/wpa_supplicant.conf
# ========================
# This line enables the use of wpa_cli which is used by rc.wireless
# if possible (to check for successful association)
ctrl_interface=/var/run/wpa_supplicant
# By default, only root (group 0) may use wpa_cli
ctrl_interface_group=0
eapol_version=1
ap_scan=1
fast_reauth=1
#country=US

# WPA protected network, supply your own ESSID and WPAPSK here:
network={
  scan_ssid=1
  ssid="nest"
  key_mgmt=WPA-PSK
  psk="secret passphrase"
}
```

The block of text we're interested in is the network block enclosed by curly braces. Here we have set the ssid for the network *"nest"*, and *"secret passphrase"* as the PSK to be used. At this point, WPA2 is properly configured. You can run ***wpa_supplicant*** and then obtain an IP address via DHCP or set a static address. Of course, this is a lot of work; there must be an easier way to do this.

rc.inet1.conf revisited

Welcome back to `rc.inet1.conf`. You're recall that in networking we used this configuration file to automatically configure NICs whenever Slackware boots. Now, we will use it to configure wifi as well.

```
  If you're using WPA2, you'll still need to setup
  wpa_supplicant.conf properly first, however.
```

Recall that each NIC had a name or number that identified the variables that correspond with it? The same hold true for wifi NICs, only they have even more variables due to the added complexity of wireless networking.

```
# rc.inet1.conf (excert)
# ======================
## Example config information for wlan0.  Uncomment the lines you need and
fill
## in your info.  (You may not need all of these for your wireless network)
IFNAME[4]="wlan0"
IPADDR[4]=""
NETMASK[4]=""
USE_DHCP[4]="yes"
```

```
#DHCP_HOSTNAME[4]="icculus-wireless"
#DHCP_KEEPRESOLV[4]="yes"
#DHCP_KEEPNTP[4]="yes"
#DHCP_KEEPGW[4]="yes"
#DHCP_IPADDR[4]=""
WLAN_ESSID[4]="nest"
#WLAN_MODE[4]=Managed
#WLAN_RATE[4]="54M auto"
#WLAN_CHANNEL[4]="auto"
#WLAN_KEY[4]="D5AD1F04ACF048EC2D0B1C80C7"
#WLAN_IWPRIV[4]="set AuthMode=WPAPSK | \
#   set EncrypType=TKIP | \
#   set
WPAPSK=96389dc66eaf7e6efd5b5523ae43c7925ff4df2f8b7099495192d44a774fda16"
WLAN_WPA[4]="wpa_supplicant"
#WLAN_WPADRIVER[4]="ndiswrapper"
```

When we discussed wired ethernet, each *n* in the variable corresponded with the *n* in *eth**n***. Here however, that no longer holds true. Notice that the variable IFNAME[4] has a value of *wlan0*. It is common for wireless cards to have an interface name other than *ethn* and that is reflected here. When `rc.inet1.conf` is read by the start-up scripts, Slackware knows to apply all these options to the *wlan0* wifi NIC instead of the (probably non-existant) eth4 wired NIC. Many of the other options are the same. IP address information is added in exactly the same way we discussed for wired network cards in networking; however, we have a lot of new variables that need some explanation.

To begin, *WLAN_ESSID[n]* and *WLAN_CHANNEL[n]* should be self-explainatory by now; they refer the the essid and frequency to use. *WLAN_MODE[n]* is either **managed** or **ad-hoc**. Anyone connecting to an access point will want to use managed mode. *WLAN_KEY[n]* is the WEP key to use, if you're forced to use WEP. *WLAN_IWPRIV[n]* is a very complicated variable that sets other variables inside itself. *WLAN_IWPRIV[n]* is used for WPA2 networks. Here you tell Slackware what authentication mode, encryption type, and key to use for WPA2 connections. Please note that *WLAN_KEY[n]* and *WLAN_IWPRIV[n]* are mutually exclusive; you can't use both on the same interface. If you successfully configure all this, then Slackware will attempt to connect to your wireless network as soon as the system boots.

But wait, that's so much work! And what if I need to connect to multiple wireless networks? I take my laptop to work and school and need to seemlessly setup those wireless connections as soon as one is within range. Doing things this way is simply too much work. You're absolutely correct.

Wicd

Introducing ***wicd***(8), the premier wired and wireless network connection manager for the laptop user on the go. Pronounced *“wicked”*, ***wicd*** is capable of storing information for any number of wireless networks you need and connecting to them with a simple command or the click of a mouse. ***wicd*** is not part of the default Slackware installation at this time, as it interferes somewhat with the normal way of configuring network adapters, but you can find it in the `/extra` directory of your Slackware install disks or at your favorite mirror. ***wicd*** is both a network connection daemon and a graphical application for configuring networks. The CLI isn't forgotten either, as ***wicd-curses***(8) is every bit as powerful as the traditional GUI front-end. In order to use ***wicd***, you will need to disable support for

any interfaces you have in `rc.inet1.conf` first.

```
# rc.inet1.conf
# =============
# Config information for eth0:
IPADDR[0]=""
NETMASK[0]=""
USE_DHCP[0]="no"
DHCP_HOSTNAME[0]=""
# Default gateway IP address:
GATEWAY=""
```

Now we can install ***wicd***, setup the daemon to run on system boot-up, and begin using a more friendly application.

```
darkstar:~# installpkg /path/to/extra/wicd/wicd-1.6.2.1-1.txz
darkstar:~# chmod +x /etc/rc.d/rc.wicd
darkstar:~# /etc/rc.d/rc.wicd start
```

If you're predominately using the console, simply run ***wicd-curses*** from your command line. If instead, you are using a graphical desktop provided by ***X***, you can start the graphical front-end from either the KDE or XFCE menu.

Optionally, you could manually run ***wicd-client***(1) from a terminal or ***run dialogue***.

On the graphical front-end, options for different networks are available via the **Preferences** button adjacent to the ESSID listed. In the terminal client, the same options can be reached by highlighting the ESSID you wish to use and pressing the right arrow key, which opens a configuration page for that network.

Chapter Navigation

Previous Chapter: Networking

Next Chapter: Basic Networking Utilities

Sources

- Original source: http://www.slackbook.org/beta
- Originally written by Alan Hicks, Chris Lumens, David Cantrell, Logan Johnson

slackbook, wireless, iwconfig, wicd