

Lo hice y lo entendí

El blog de Vicente Navarro

Configurar WEP y WPA en línea de comandos y en el arranque en Debian y Ubuntu

Desde que el NetworkManager se ocupa de las conexiones de red en nuestras distribuciones, la antaño ardua tarea de configurar la red en Linux se ha simplificado ahora enormemente. Sin embargo, puede haber muchos casos en los que sigamos necesitando configurar la red en la consola. También hay muchos casos en los que es imprescindible que un sistema se conecte automáticamente a una red WiFi durante el arranque de la misma.

En esta entrada veremos cómo conseguir configurar nuestra distribución Debian o Ubuntu para conectarnos a una red WiFi encriptada con WEP o con WPA usando la línea de comandos. También veremos cómo modificar el fichero `/etc/network/interfaces` para que el sistema se conecte automáticamente a una red WiFi durante el arranque.

Para que no interfiera con mis pruebas, he deshabilitado el NetworkManager antes de comenzar con las mismas (lo que causará que nos desconectemos de las redes que éste gestione):

```
$ sudo /etc/init.d/NetworkManager stop
```

También he verificado si el demonio `wpa_supplicant` está corriendo en segundo plano y lo he matado con un simple `kill`. Hablaremos más adelante de este proceso.

Configuración de WEP

Lo primero que necesitamos comprobar es si el sistema ha cargado correctamente el driver para nuestro interfaz WiFi y, por tanto, éste aparece en la salida del comando `iwconfig`. `iwconfig` es parte de las Wireless Tools for Linux y nos mostrará los interfaces que soporten las extensiones para WiFi de Linux:

```
$ iwconfig
lo          no wireless extensions.

eth1        no wireless extensions.

eth0        no wireless extensions.

pan0        no wireless extensions.

wmaster0    no wireless extensions.

wlan0       IEEE 802.11bg  ESSID:""
            Mode:Managed  Frequency:2.412 GHz  Access Point: Not-Associated
            Tx-Power=0 dBm
            Retry min limit:7   RTS thr:off   Fragment thr=2352 B
            Power Management:off
            Link Quality:0  Signal level:0  Noise level:0
            Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
            Tx excessive retries:0  Invalid misc:0  Missed beacon:0
```

Hay chipsets WiFi que pueden necesitar que utilicemos herramientas propias para crear un interfaz con extensiones wireless. Por ejemplo, con los chipsets Atheros, que usan el driver MadWifi, por defecto tendremos un interfaz maestro `wifi0` sin extensiones wireless que nos permite crear múltiples interfaces `athX` para diversas necesidades. Con el comando `wlanconfig`, incluido en Debian/Ubuntu en el paquete `madwifi-tools`, podríamos crear un interfaz normal (en modo “estación”), con el siguiente comando:

```
wlanconfig ath0 create wlandev wifi0 wlanmode sta
```

Con el driver de chipsets Ralink pasa algo parecido: `Rt2x00 README: Interfaces`.

Sin embargo, la mayoría de los drivers, aunque se basen en un interfaz maestro, por defecto también nos crean un interfaz estándar con extensiones wireless que podremos ver con `iwconfig` y que será el que usemos de ahora en adelante.

En la salida del `ifconfig`, el interfaz no nos sale, ya que está *down*:

```
$ ifconfig
lo          Link encap:Local Loopback
            inet addr:127.0.0.1  Mask:255.0.0.0
            inet6 addr: ::1/128  Scope:Host
            UP LOOPBACK RUNNING  MTU:16436  Metric:1
            RX packets:100 errors:0 dropped:0 overruns:0 frame:0
            TX packets:100 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:0
            RX bytes:7448 (7.4 KB)  TX bytes:7448 (7.4 KB)
```

Para configurar ahora la conexión a una red WiFi encriptada con WEP cuyo ESSID es “ssidpruebas” y cuya clave WEP de 104 bits en hexadecimal es “11223344556677889900aabbcc”, haremos:

```
$ sudo iwconfig wlan0 essid ssidpruebas key 1122-3344-5566-7788-9900-aabb-cc
```

El comando `iwconfig` tiene muchas más opciones para configurar otros parámetros o para pasarle la contraseña en formato texto (lo haríamos con “`key s:clavedificil`”). Su página de man es la mejor referencia para consultarlas.

Pocos segundos más tarde ya podremos comprobar que el interfaz de red se ha asociado al Access Point (AP):

```
$ iwconfig
lo          no wireless extensions.

eth1        no wireless extensions.
```

```

eth0      no wireless extensions.

pan0      no wireless extensions.

wmaster0  no wireless extensions.

wlan0     IEEE 802.11bg  ESSID:"ssidpruebas"
          Mode:Managed  Frequency:2.462 GHz  Access Point: 00:2A:3B:24:1A:ED
          Bit Rate=1 Mb/s   Tx-Power=23 dBm
          Retry min limit:7   RTS thr:off   Fragment thr=2352 B
          Power Management:off
          Link Quality=67/100  Signal level:-61 dBm
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:0  Invalid misc:0  Missed beacon:0

```

En una red cableada, esto sería equivalente a conectar el cable con conector RJ45 al interfaz de red. Tenemos acceso a nivel 2 pero la red IP aún no está configurada.

Ahora levantamos el interfaz con `ifconfig` y obtenemos la configuración de IP con el cliente de DHCP, aunque también podríamos asignarle una IP fija:

```

$ sudo ifconfig wlan0 up

$ sudo dhclient3 wlan0
There is already a pid file /var/run/dhclient.pid with pid 11426
killed old client process, removed PID file
Internet Systems Consortium DHCP Client V3.1.1
Copyright 2004-2008 Internet Systems Consortium.
All rights reserved.
For info, please visit http://www.isc.org/sw/dhcp/

Listening on LPF/wlan0/10:90:6a:37:ae:cc
Sending on   LPF/wlan0/10:90:6a:37:ae:cc
Sending on   Socket/fallback
DHCPREQUEST of 192.168.4.4 on wlan0 to 255.255.255.255 port 67
DHCPACK of 192.168.4.4 from 192.168.4.1
bound to 192.168.4.4 -- renewal in 104602 seconds.

```

Si queremos que esto mismo se haga automáticamente durante el arranque de la máquina, hemos de seguir las indicaciones del documento `/usr/share/doc/wireless-tools/README.Debian`, que nos dice que las entradas en el fichero `/etc/network/interfaces` del tipo:

```
wireless-[function] [value]
```

se traducen en comandos `iwconfig`:

```
iwconfig [interface] [function] [value]
```

Por tanto, las siguientes líneas en el fichero `/etc/network/interfaces` conseguirán que el interfaz se conecte a la red WEP automáticamente durante el arranque:

```

auto wlan0
iface wlan0 inet dhcp
    wireless-essid ssidpruebas
    wireless-key 1122-3344-5566-7788-9900-aabb-cc

```

También lo podemos probar sin reiniciar:

```

$ sudo /etc/init.d/networking stop
* Deconfiguring network interfaces... [ OK ]

$ sudo /etc/init.d/networking start
* Configuring network interfaces...
Internet Systems Consortium DHCP Client V3.1.1
Copyright 2004-2008 Internet Systems Consortium.
All rights reserved.
For info, please visit http://www.isc.org/sw/dhcp/

Listening on LPF/wlan0/10:90:6a:37:ae:cc
Sending on   LPF/wlan0/10:90:6a:37:ae:cc
Sending on   Socket/fallback
DHCPDISCOVER of 192.168.4.4 on wlan0 to 255.255.255.255 port 67 interval 7
DHCPOFFER of 192.168.4.4 from 192.168.4.1
DHCPREQUEST of 192.168.4.4 on wlan0 to 255.255.255.255 port 67
DHCPACK of 192.168.4.4 from 192.168.4.1
bound to 192.168.4.4 -- renewal in 126729 seconds.

```

Configuración de WPA

Las Wireless Tools for Linux no sirven para configurar interfaces WiFi para usar redes WPA. Por tanto, para WPA no podemos usar `iwconfig` como hemos hecho con WEP.

Hay drivers, como el de Ralink que proporcionan una forma propietaria, a través de comandos `iwpriv`, de configurar WPA en un interfaz: Debian rt2500 Howto: Using WPA.

Sin embargo, la forma estándar de hacerlo es mediante el demonio `wpa_supplicant` (`wpa_supplicant` README).

El demonio del NetworkManager levanta automáticamente el `wpa_supplicant`, de modo que para seguir con las pruebas, es buena idea parar el NetworkManager y matar cualquier proceso `wpa_supplicant` que pueda quedar activo, como hemos comentado al principio.

Partimos, de nuevo, con un interfaz con extensiones wireless sin configurar:

```

$ iwconfig
lo      no wireless extensions.

```

```

eth1      no wireless extensions.

eth0      no wireless extensions.

pan0      no wireless extensions.

wmaster0  no wireless extensions.

wlan0     IEEE 802.11bg  ESSID:""
          Mode:Managed  Frequency:2.412 GHz  Access Point: Not-Associated
          Tx-Power=0 dBm
          Retry min limit:7   RTS thr:off   Fragment thr=2352 B
          Power Management:off
          Link Quality:0  Signal level:0  Noise level:0
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:0  Invalid misc:0  Missed beacon:0

```

Editamos un fichero de configuración (por ejemplo en `/etc/wpa_supplicant/wpa_supplicant.conf`) como el siguiente en el que especificaremos el tipo de clave (WPA-PSK), el nombre del SSID (`ssidpruebas`) y la contraseña (`112233445566aabb`):

```

ctrl_interface=/var/run/wpa_supplicant
network={
    ssid="ssidpruebas"
    key_mgmt=WPA-PSK
    psk="112233445566aabb"
}

```

En el README encontraremos otros ejemplos que muestran diferentes configuraciones, pero ésta sin duda es la más estándar para un entorno doméstico.

Ahora, si arrancamos el demonio `wpa_supplicant` (en el siguiente ejemplo no usamos la opción `-B` para que se quede funcionando en segundo plano puesto que nos interesa ver la salida en el terminal, pero sería lo más normal usarla):

```

$ sudo wpa_supplicant -c/etc/wpa_supplicant/wpa_supplicant.conf -iwlan0
CTRL-EVENT-SCAN-RESULTS
CTRL-EVENT-SCAN-RESULTS
Trying to associate with 10:2a:3b:24:1a:ed (SSID='ssidpruebas' freq=2462 MHz)
Associated with 10:2a:3b:24:1a:ed
WPA: Key negotiation completed with 10:2a:3b:24:1a:ed [PTK=TKIP GTK=TKIP]
CTRL-EVENT-CONNECTED - Connection to 10:2a:3b:24:1a:ed completed (auth) [id=0 id_str=]

```

Vemos que el interfaz se asocia sin problemas al punto de acceso:

```

$ iwconfig
lo      no wireless extensions.

eth1    no wireless extensions.

eth0    no wireless extensions.

pan0    no wireless extensions.

wmaster0 no wireless extensions.

wlan0   IEEE 802.11bg  ESSID:"ssidpruebas"
        Mode:Managed  Frequency:2.462 GHz  Access Point: 10:2A:3B:24:1A:ED
        Bit Rate=48 Mb/s   Tx-Power=23 dBm
        Retry min limit:7   RTS thr:off   Fragment thr=2352 B
        Power Management:off
        Link Quality=30/100  Signal level:-66 dBm
        Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
        Tx excessive retries:0  Invalid misc:0  Missed beacon:0

```

Así que ahora ya podemos levantar el interfaz y pedir una IP por DHCP (aunque, como siempre, le podríamos poner una fija):

```

$ sudo ifconfig wlan0 up

$ sudo dhclient3 wlan0
There is already a pid file /var/run/dhclient.pid with pid 14066
killed old client process, removed PID file
Internet Systems Consortium DHCP Client V3.1.1
Copyright 2004-2008 Internet Systems Consortium.
All rights reserved.
For info, please visit http://www.isc.org/sw/dhcp/

Listening on LPF/wlan0/10:90:6a:37:ae:cc
Sending on   LPF/wlan0/10:90:6a:37:ae:cc
Sending on   Socket/fallback
DHCPREQUEST of 192.168.4.4 on wlan0 to 255.255.255.255 port 67
DHCPACK of 192.168.4.4 from 192.168.4.1
bound to 192.168.4.4 -- renewal in 102633 seconds.

```

Para hacer esto mismo automáticamente durante el arranque seguiremos las indicaciones del documento

`/usr/share/doc/wpa_supplicant/README.Debian.gz` y añadiremos unas líneas a `/etc/network/interfaces` como las siguientes:

```

auto wlan0
iface wlan0 inet dhcp
    wpa-ssid ssidpruebas
    wpa-psk 112233445566aabb

```

Tras lo cual podremos comprobar que el interfaz se conecta automáticamente a la red WiFi durante el arranque del sistema:

```

$ sudo /etc/init.d/networking stop
* Deconfiguring network interfaces...

```

```
$ sudo /etc/init.d/networking start
* Configuring network interfaces...
Internet Systems Consortium DHCP Client V3.1.1
Copyright 2004-2008 Internet Systems Consortium.
All rights reserved.
For info, please visit http://www.isc.org/sw/dhcp/

Listening on LPF/wlan0/10:90:6a:37:ae:cc
Sending on LPF/wlan0/10:90:6a:37:ae:cc
Sending on Socket/fallback
DHCPDISCOVER on wlan0 to 255.255.255.255 port 67 interval 6
DHCPOFFER of 192.168.4.4 from 192.168.4.1
DHCPREQUEST of 192.168.4.4 on wlan0 to 255.255.255.255 port 67
DHCPACK of 192.168.4.4 from 192.168.4.1
bound to 192.168.4.4 -- renewal in 105559 seconds.
```

Quedándose el demonio wpa_supplicant funcionando en segundo plano:

```
$ ps -ef | grep wpa
root      15184      1    0 10:18 ?           00:00:00 /sbin/wpa_supplicant -B -P /var/run/wpa_supplicant.wlan0.pid -i wlan0 -D wext -q
```

:wq

Entradas relacionadas

- ¿Aún sigues con encriptación WEP para tu WiFi? (4)
- vga=ask y los modos VESA disponibles en el sistema (3)
- Ubuntu y Debian eliminan ms-sys de sus repositorios (3)
- sysstat, la colección de herramientas de monitorización de rendimiento (3)
- Midiendo el ancho de banda de red con IPerf (y con scp, netcat, wget) (3)
- Linux no mata discos duros, se mueren solos (3)

Esta entrada fue publicada bajo una licencia CC el 1 marzo 2009 a las 12:26 y pertenece a la categoría GNU/Linux, Redes. Etiquetas: Debian, GNU/Linux, seguridad, Ubuntu, WEP, WiFi, WPA. Puedes seguir las respuestas a través de RSS. Ya no se pueden dejar comentarios, pero puedes hacer trackback desde tu página.