Anexo 3

INDICE

1. Algunos enrutadores inalámbricos basados en Software Libre 2

2. Chipsets usados en tarjetas inalámbricas. Algunas marcas y funcionamiento general. 2

3. Ventajas de las soluciones inalámbricas basadas en Software Libre 3

4. Comandos de configuración de tarjetas inalámbricas con Software Libre 3

5. Métodos de configuración de enrutamiento en enrutadores con Software Libre..... 7

REFERENCIAS 8

1. Algunos enrutadores inalámbricos basados en Software Libre

Los enrutadores basados en software libre ofrecen funciones para operar como *bridge*, punto de acceso y router simultáneamente, utilizan bajo consumo de energía entre 4 y 5watts, generalmente vienen en una tarjeta madre, en la cual se pueden conectar tarjetas inalámbricas como pueden ser las de *chipset Prism* que son las más utilizadas por los usuarios de Linux, así mismo utilizan una memoria la cual puede ser una *SD, Standard Disk o Compact Flash*, la cual tiene una ranura especial para ser insertada, en esta memoria se instala el sistema operativo. Utilizan también un procesador, el cual puede ser un AMD.

Los enrutadores más conocidos son los de placa *Soekris* NET4521 y NET4511, están basados en un procesador x86 AMD, y utilizan una *Compact Flash*, y utilizan las tarjetas inalámbricas *Atheros* o *Prism*, las cuales pueden ser PCI o *miniPCI*.

2. Chipsets usados en tarjetas inalámbricas. Algunas marcas y funcionamiento general.

El "***chipset***" es el conjunto (*set*) de chips que se encargan de controlar determinadas funciones de un dispositivo, en este caso la tarjeta inalámbrica

En el mercado tenemos varios tipos de *chipset*, entre ellos: Chipset Prism, Chipset Hermes, Chipset Symbol, Chipset Atheros, Chipser Broadcom.

Chipset Prism

Este es uno de los más utilizados por la comunidad GNU/Linux, y ya que toda la información referente a estos se puede conseguir fácilmente, eso ha conllevado que se creen diversos controladores para este chipset como son las herramientas linux-wlan-ng, controladores Prism54, Airjack , controladores HostAP.

En cuanto a su funcionamiento tiene un controlador de MAC que realiza la mayor parte de las operaciones básicas del protocolo 802.11, posee un motor WEP que agiliza el trabajo de criptografía.

Chipset Aironet

A Partir del *chipset Prism Cisco* añadió algunas características como permitir el paso de la banda ISM a otra frecuencia y control de potencia, sin embargo con el tiempo estos dos *chipset* sean diferentes, este *chipset* tiene una excelente sensibilidad a la recepción y permite monitorización.

Chipset Hermes

Este *chipset* fue desarrollado por *Lucent*, es de código cerrado, aun que *Lucent* libero parte del código fuente para las funciones básicas del controlado Orinoco. Este *chipset* identifica decide el *essid* ni bien se activa la interfaz inalámbrica, posee el modo monitor aun que es necesario añadir un parche.

3. Ventajas de las soluciones inalámbricas basadas en Software Libre

La principal ventaja es que se reducen los costos ya que no se necesita pagar por licencia, gracias a la licencia GNU/Linux, y por lo mismo que se puede acceder al código fuente lo que permite, que nuevos usuarios puedan estudiar el software, copiarlo y modificarlo, redistribuirlos, y mejorarlo.
De esta manera se consigue que los demás usuarios cooperen para que se mejore el software, y se puedan dividir las tareas para arreglar los *bugs*. Así se crean nuevas mejoras como es el caso del *chipset Prism*, el cual tiene varios controladores.
Tienen más servicios como es el caso de Linux AP que ofrece *telnet, httpd* y ha mejorado las opciones iniciales del punto de acceso comercial como son modo master, cliente, repetidor, *bridge*, WDS, *aodv, openvpn*.

4. Comandos de configuración de tarjetas inalámbricas con Software Libre

Para configurar una tarjeta inalámbrica es necesario asegurarnos que la tarjeta esta siendo reconocida por la interfaz, para eso utilizamos el comando *iwconfig*.

```
isaac@isaac-laptop:~$ iwconfig
lo no wireless extensions.

eth0 no wireless extensions.

eth1 IEEE 802.11b/g ESSID:"wifipucp" Nickname:"Broadcom 4318"
Mode:Managed Frequency=2.484 GHz Access Point: Invalid
Bit Rate=1 Mb/s Tx-Power=18 dBm
RTS thr:off Fragment thr: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

sit0 no wireless extensions.

isaac@isaac-laptop:
```

Como vemos en este *host* la interfaz eth1 ha sido reconocida como interfaz inalámbrica, en caso no fuera reconocida aparecería como *no wireless extensions* como es el caso de las demás interfaces.

Sin embargo, la tarea de hacer reconocer una tarjeta inalámbrica no es tan sencilla, y algunos casos puede resultar un dolor de cabeza para muchos.

El problema es que muchos fabricantes tienen convenios con Windows y producen sus controladores solo para este sistema operativo, por lo que existe algunos grupos de desarrolladores que programan controladores compatibles para Linux.

Uno de las aplicaciones más conocidas es el *ndiswrapper,* el cual es una herramienta que nos ayuda a reconocer varios tipos de tarjetas inalámbricas, haciendo uso del controlador que es utilizado en Windows.

Para el caso a ejemplificar es una tarjeta inalámbrica *Dell Wireless MiniPCI 1370* con *chipset Broadcom Corporation BCM4318 [AirForce One 54g] 802.11g Wireless LAN Controller (rev 02),* para poder ver el tipo de *chipset* de nuestra tarjeta utilizamos el comando

```
$lspci
```

Descargar e instar el *ndiswrapper*, si estamos usando Ubuntu podemos descargarlo de los repositorios utilizando el Synaptic o bien con el comando.

```
$sudo apt-get install ndiswrapper-utils
```

Si no se consiguiese instalar, revisar los repositorios, de ser correcta la instalación pasamos a descargar los controladores de Windows, se recomienda descargar el controlador y no copiar y pegar el mismo que esta siendo utilizado por Windows.

Debemos tener los archivos bcmwl5.inf, y bcmwl5.sys.

Bien luego de ello utilizamos el siguiente comando

```
$sudo ndiswrapper -i bcmwl5.inf
```

(Ubicarse en la carpeta donde se encuentra el *driver*)

De ser correcto este procedimiento utilizamos el siguiente comando

```
$sudo ndiswrapper -l
```

y nos debería salir:

```
Installed ndis drivers:
bcmwl5 driver present, hardware present
```

Ahora levantamos el modulo del *ndiswrapper*, con el comando.

```
$sudo modprobe ndiswrapper
$sudo ndiswrapper -m
```

Verificamos que se haya levantado correctamente con el comando

```
$lsmod | grep ndiswrapper
```

Si todo ha sucedido con éxito el LED indicador de la tarjeta inalámbrica debería haberse encendido. Si queremos utilizar el *ndiswrapper* nos faltaría hacer la configuración de la tarjeta que se detalla más adelante.

Aunque es muy útil el *ndiswrapper* ya que solamente nos proporciona el uso cotidiano de la tarjeta inalámbrica, sin embargo si queremos realizar tareas de auditoria en redes inalámbricas, se recomiendo no usar *ndiswrapper,* ya que no soporta aplicaciones como *Aircrack*. Por lo que es más conveniente levantar la tarjeta con el *driver* creado especialmente para las tarjetas inalámbricas del tipo *bcm43xx* o en su defecto de otra tarjeta que se este utilizando.

Para ello mejor desinstalamos el *ndiswrapper* desde el synaptic o bajamos el modulo.

Ahora debemos de descargar el paquete wl_apsta.o el cual contiene los *drivers* bcm4318 entre otros, desde **http://drinus.net/airport/wl_apsta.o** y el paquete *bcm43xx-fwcutter* eso lo podemos hacer con el comando:

```
$sudo apt-get install bcm43xx-fwcutter
```

Ahora corremos el paquete *bcm43xx-fwcutter* con el archivo descargado, buscar donde esta el archivo, y aplicar el siguiente comando

```
$sudo bcm43xx-fwcutter -w <downloaded file>
```

```
$sudo bcm43xx-fwcutter -w  wl_apsta.o
```

A continuación levantamos el modulo así como hicimos con el *ndiswrapper*

```
$sudo modprobe bcm43xx
```

Ahora verificamos nuevamente con el *iwconfig* para ver con que alias esta la interfaces, (ver la respuesta a este comando antes mencionada).

Ahora pasamos a configurar la tarjeta.
Primero debemos de ver las redes disponibles, para nuestro caso la interfaz inalámbrica se encuentra bajo el alias de eth1

```
sudo iwlist eth1 scan
```

Debe salir una respuesta como la siguiente:

```
isaac@isaac-laptop:-$ sudo iwlist eth1 scan
eth1      Scan completed :
          Cell 01 - Address: 00:90:4B:67:A9:F6
                    ESSID:"wifipucp"
                    Protocol:IEEE 802.11bg
                    Mode:Master
                    Channel:4
                    Encryption key:on
                    Bit Rates:1 Mb/s; 2 Mb/s; 5.5 Mb/s; 6 Mb/s; 9 Mb/s
                              11 Mb/s; 12 Mb/s; 18 Mb/s; 24 Mb/s; 36 Mb/s
```

```
                    48 Mb/s; 54 Mb/s
                Quality=100/100  Signal level=-54 dBm
                Extra: Last beacon: 268ms ago
        Cell 02 - Address: 00:90:4B:1C:7B:E4
                ESSID:"wifipucp"
                Protocol:IEEE 802.11bg
                Mode:Master
                Channel:9
                Encryption key:on
                Bit Rates:1 Mb/s; 2 Mb/s; 5.5 Mb/s; 6 Mb/s; 9 Mb/s
                        11 Mb/s; 12 Mb/s; 18 Mb/s; 24 Mb/s; 36 Mb/s
                        48 Mb/s; 54 Mb/s
                Quality=100/100  Signal level=-67 dBm
                Extra: Last beacon: 148ms ago
isaac@isaac-laptop:-$
```

Ahora por ejemplo si deseamos conectarnos a la red *wifipucp*, configuramos el tipo de red, el essid, una clave WEP.

Agregando el essid:

```
isaac@isaac-laptop:~$ sudo iwconfig eth1 essid 'wifipucp'
```

Ponemos en modo Infraestructura:

```
isaac@isaac-laptop:~$ sudo iwconfig eth1 mode Managed
```

Agregamos la clave WEP: **********

```
isaac@isaac-laptop:~$ sudo iwconfig eth1 key restricted **********
```

Luego de hacer cambios en la interfaz para que hagan efecto debemos reiniciarla.

```
isaac@isaac-laptop:~$ sudo ifconfig eth1 down
isaac@isaac-laptop:~$ sudo ifconfig eth1 up
```

Y como esta red utiliza DHCP, utilizamos el siguiente comando para obtener la dirección IP, mascara, DNS.
isaac@isaac-laptop:~$ sudo dhclient eth1

```
isaac@isaac-laptop:-$ sudo dhclient eth1
Internet Systems Consortium DHCP Client V3.0.4
Copyright 2004-2006 Internet Systems Consortium.
All rights reserved.
For info, please visit http://www.isc.org/sw/dhcp/

Listening on LPF/eth1/00:14:a5:50:4b:81
Sending on LPF/eth1/00:14:a5:50:4b:81
Sending on Socket/fallback
DHCPDISCOVER on eth1 to 255.255.255.255 port 67 interval 4
DHCPOFFER from 192.168.30.10
DHCPREQUEST on eth1 to 255.255.255.255 port 67
DHCPACK from 192.168.30.10
bound to 192.168.30.211 -- renewal in 1417 seconds.
isaac@isaac-laptop:-$
```

Y listo nos hemos conectado a una red, también podemos editar el archivo */etc/network/interfaces* para no configurar a cada momento.

```
isaac@isaac-laptop:~$ sudo vi /etc/network/interfaces
auto lo
iface lo inet loopback
auto eth0
iface eth0 inet dhcp
auto eth1
iface eth1 inet dhcp
wireless-essid wifipucp
wireless-key **********
auto eth2
iface eth2 inet dhcp
```

Finalmente estamos listos para utilizar los servicios de la WLAN.

5. Métodos de configuración de enrutamiento en enrutadores con Software Libre.

Enrutamiento Estático, viene por incluido en el kernel de Linux, se muestra como el paquete IPROUTE, aun que es posible obtenerlo de la siguiente dirección ftp://ftp.inr.ac.ru/ip-routing, o en su defecto de los repositorios. Este paquete nos proporciona comandos para realizar la administración del trafico IP, configurar las interfaces, protocolos, se pueden realizar también monitoreo de periféricos, y el uso de tablas de enrutamiento.

Como ejemplo mostramos la asignación de direcciones ip estáticas con el comando
$*ip address add 192.168.1.115 dev eth1*

Como también podemos establecer las rutas por ejemplo modificando las tablas de enrutamiento, acá mostramos un ejemplo utilizado en el curso de Análisis de Protocolos, donde modificamos el archivo */etc/iproute2/rt_tables* agregando tablas, para luego asignarles reglas de enrutamiento.

```
# reserved values
#
#255    local
#254    main
#253    default
#0      unspec
#
# local
#
#1      inr.ruhep
100     ISP1
200     ISP2
```

A las cuales se les aplica comandos de este tipo, para la tabla ISP1 por ejemplo.

ip route add 192.168.1.0/24 dev eth1 src 192.168.1.155 table isp1
ip route add default via 192.168.1.2 table isp

Enrutamiento Dinámico, de manera similar a un router en Linux podemos implementar métodos de enrutamiento dinámico como RIP, OSPF.

Ejemplo de configuración de RIP:

Ejecutamos el demonio con /usr/sbin/router luego creamos el siguiente script y lo ejecutamos.

```
!/sbin/bash

# Habilitamos la redirección de paquetes
echo 1 > /proc/sys/net/ipv4/ip_forward

# Configuramos las interfaces
/sbin/ifconfig eth0 192.168.1.2 netmask 255.255.255.0
/sbin/ifconfig eth1 192.168.2.1 netmask 255.255.255.0
/sbin/ifconfig eth2 192.168.3.1 netmask 255.255.255.0

# Gateway por defecto
/sbin/route add default gw 192.168.1.1
# Arrancamos routed
/usr/sbin/routed
```

REFERENCIAS

[1]. Using Broadcom Wireless in Ubuntu Edgy 6.10: https://help.ubuntu.com/community/WifiDocs/Driver/bcm43xx/Edgy
[2]. Ndiswrapper: http://ndiswrapper.sourceforge.net/mediawiki/index.php/Installation
[3]. El chipset:http://www.conozcasuhardware.com/quees/chipset.htm
[4]. Aironet : http://www.cisco.com/en/US/products/hw/wireless/ps430/prod_technical_reference_list.html
[5]. Muñoz J. ,Material del Curso Análisis de Protocolos, PUCP
[6]. Chávez A. ,Material del Curso Ingeniería Inalámbrica, PUCP