Redes y Seguridad

Comando *ifconfig*

Mediante este comando nosotros vamos a poder configurar la red momentáneamente es decir cuando reiniciamos la computadora la configuración que realizamos con ***ifconfig*** se pierde.

Como dijimos en muchas oportunidades la interfaz ***Ethernet*** es ***eth[0..n]***, como también otras interfaz como se muestra en la siguiente tabla :

Tipo de interfaz	Nombre
Ethernet	eth0, eth1,...
Loopback	lo
Modem	ppp0, ppp1,...
Wi-Fi	wlan0, wlan1,...
BlueTooth	hci0, hci1,...
Token Ring	tr0, tr1,...

ifconfig [PARAMETROS] <interfaz> [<dirección>] [PARAMETROS]

Para ver la configuración de la placas simplemente escribimos ***ifconfig*** :

$ ifconfig

eth0 Link encap:Ethernet HWaddr 48:5b:39:95:38:ff
inet addr:192.168.0.101 Bcast:192.168.0.255 Mask:255.255.255.0
inet6 addr: fe80::4a5b:39ff:fe95:38ff/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:811944 errors:0 dropped:0 overruns:0 frame:0
TX packets:607144 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:1112779483 (1.0 GiB) TX bytes:48286511 (46.0 MiB)
Interrupt:27

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:1122 errors:0 dropped:0 overruns:0 frame:0
TX packets:1122 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:109049 (106.4 KiB) TX bytes:109049 (106.4 KiB)

Opciones de ***ifconfig*** :

Opciones	Descripción
-a	Muestra todas las interfaces disponibles.
-m	Muestra una lista corta con las interfaces.
-v	Muestra más descripción cuando es necesaria.
<interfaz>	Nombre de la interfaz.
<dirección>	Dirección que será asignada a la interfaz.
<HW>	Tipo de hardware asignado a la interfaz.
<AF>	Tipo de dirección que será asignada a la interfaz.
up	indica que la interfaz estará activa.
down	indica que la interfaz estará desactivada.

Si queremos ver una placa determinada por ejemplo ***eth0*** :

$ ifconfig eth0

eth0 Link encap:Ethernet HWaddr 48:5b:39:95:38:ff
inet addr:192.168.0.101 Bcast:192.168.0.255 Mask:255.255.255.0
inet6 addr: fe80::4a5b:39ff:fe95:38ff/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:811944 errors:0 dropped:0 overruns:0 frame:0
TX packets:607144 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:1112779483 (1.0 GiB) TX bytes:48286511 (46.0 MiB)
Interrupt:27

Si queremos ver todas las placas ***ifconfig -a*** esto incluye las placas no configuradas.

Si queremos deshabilitar la placa ***eth0*** :

$ ifconfig eth0 down

Si queremos habilitarla :

$ ifconfig eth0 up

Si queremos asignas a la interfaz de red ***eth0*** la ip ***192.168.1.1*** netmask ***255.255.255.0*** y broadcast ***192.168.1.255***.

$ ifconfig eth0 192.168.1.1 netmask 255.255.255.0 broadcast 192.168.1.255

Si queremos cambiar el ***MTU*** (unidad máxima de transmisión) , para ***ethernet*** el tamaño máximo de un paquete a transmitir por transacción en TCP/IP por defecto es de ***1500 bytes***.

$ ifconfig eth0 mtu XX

Para poner en modo promiscuo la placa :

$ ifconfig eth0 promisc
$ ifconfig eth0

eth0 Link encap:Ethernet HWaddr 48:5b:39:95:38:ff
inet addr:192.168.0.101 Bcast:192.168.0.255 Mask:255.255.255.0
inet6 addr: fe80::4a5b:39ff:fe95:38ff/64 Scope:Link
*UP BROADCAST RUNNING **PROMISC** MULTICAST MTU:1500 Metric:1*
RX packets:832501 errors:0 dropped:0 overruns:0 frame:0
TX packets:620578 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:1137859281 (1.0 GiB) TX bytes:49367986 (47.0 MiB)
Interrupt:27

Para sacar el modo promiscuo la placa :

$ ifconfig eth0 -promisc

Para crear una placa de red virtual :

$ ifconfig eth0:0 192.168.1.2

MAC Spoofing

Como cambiar la dirección MAC del dispositivo por una diferente en una interfaz de red llamada eth0:

$ ifconfig eth0 down
$ ifconfig eth0 hw ether 00:11:22:33:44:55 up

Dispositivos de red Inalámbricos (wireless devices)

Para usar las herramientas de wireless tenemos que instalar el paquete ***wireless-tools.***

$ apt-get install wireless-tools

El comando **iwconfig** nos permite listar los dispositivos que tienen soporte inalámbrico:

$ iwconfig

lo no wireless extensions.

eth0 no wireless extensions.

wlan0 IEEE 802.11bg ESSID:"SpeedyWiFi"
Mode:Managed Frequency:2.412 GHz Access Point: Not-Associated
Tx-Power=20 dBm
Retry long limit:7 RTS thr:off Fragment thr:off
Encryption key:off
Power Management:off

pan0 no wireless extensions.

Tenemos también el comando ***iwlist*** que entre otras cosas nos permite scanear las redes inalámbricas.

iwlist [interface] opciones

Opciones de ***iwlist*** :

Opciones	Descripción
scan	Nos mostrará información de todas las redes inalámbricas que nuestra tarjeta detecta. Obviamente en modo monitor dará cero resultados.
frequency	Nos mostrara los diferentes valores de frecuencia y su correspondencia en el numero de canal validos para nuestra tarjeta así como la frecuencia y el canal en el que se encuentra en esos momentos la tarjeta.
channel	Igual que el anterior.
rate	Indica la velocidad de comunicación que nuestra tarjeta soporta así como la velocidad actual (mediante current bit rate).

$ iwlist wlan0 scanning

…

Dispositivos de red Bluetooth

Bluetooth también permite crear redes WPAN (Wireless Personal Area Network, Red de Área Personal Inalámbrica).

$ apt-get install bluetooth

Listar dispositivo ***Bluetooth*** soportados e instalados en el sistema.

$ hciconfig

Comprobar que detecta el interfaz Bluetooth:

$ hciconfig dev

Buscar dispositivos remotos (obtenemos la MAC del dispositivo):

$ hciconfig scan

Investigar dispositivos remotos:

$ hciconfig inq

Configuración de los DNS

Para establecer los servidores DNS que se encargarán de resolver los nombres, se debe editar el archivo ***/etc/resolv.conf***

Este archivo debe contener la definición de los servidores, de la siguiente forma

$ vi /etc/resolv.conf

```
nameserver    200.74.121.11
nameserver    190.160.0.11
```

Definición de nombres particulares

Editando el archivo **/etc/hosts** es posible asignar un nombre a una dirección específica.

$ more /etc/hosts

```
127.0.0.1       localhost
127.0.1.1       urano
# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
ff02::3 ip6-allhosts
```

Si queremos por ejemplo agregar la maquina de Juan con ip 192.168.1.100 hacemos lo siguiente :

$ echo "192.168.1.100 juan" >> /etc/hosts

Y ahora ya podemos referirnos por el nombre en vez de la ip.

$ ping juan

```
PING juan (192.168.1.100) 56(84) bytes of data.
64 bytes from juan (192.168.1.100): icmp_seq=1 ttl=64 time=2.63 ms
64 bytes from juan (192.168.1.100): icmp_seq=2 ttl=64 time=3.58 ms
^C
```

Nombre del host

En el archivo ***/etc/hostname*** tenemos el nombre del equipo :

$ more /etc/hostname

centrux

Enrutamiento

Configuración de la puerta de enlace ejemplo de como cambiar la puerta de enlace mediante el comando **route.**

$ route

```
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.0.0     *               255.255.255.0   U     0      0        0 eth0
default         192.168.0.1     0.0.0.0         UG    0      0        0 eth0
```

Si queremos borrar la ruta por defecto el ***gateway*** realizamos lo siguiente :

$ route del default

Para agregar la nueva ruta por defecto :

$ route add default gw 192.168.1.101

El cliente DHCP

DHCP significa Protocolo de Configuración Dinámica del Anfitrión (Dynamic Host Configuration Protocol). Es un protocolo muy usado para proporcionar automáticamente información como direcciones IP, máscaras de subred e información de encaminamiento (enrutamientos) entre computadores.
Si nuestra red usa DHCP, necesitarás un cliente DHCP para poder conectarte a ella mediante este protocolo. DHCP también se usa en la mayoría de las conexiones a internet mediante módems.

El cliente ***dhcp dhclient***

dhclient es un client dhcp, distribuido en varias distribuciones GNU/Linux.

$ dhclient eth0

Configuración del archivo de interfaces

A diferencia del comando ***ifconfig*** en el archivo de interfaz cada vez que reiniciamos el equipo este levantara la configuración de este archivo.

$ vi /etc/network/interfaces

```
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug eth0
iface eth0 inet dhcp
```

Como vemos la primer interfaz que encontramos es ***loopback*** (lo).

```
auto lo
iface lo inet loopback
```

- ***auto lo :*** se encarga de levantar la interfaz que se especifica.
- ***iface*** : diminutivo de interfaz.
- ***inet*** : configura dicha interfaz para valores ipv4 (***inet6***, para las direcciones ipv6).
- ***loopback*** : especifica que se debe cargar la configuración de ***loopback***.

Configuración Estática

Inicialmente comentar que está configurado para que no se levanten las interfaces, sino para hacerlo manualmente. En el caso de querer tener una interfaz que se levante por defecto cuando se arranca el pc deberemos añadir ***auto interfaz***. Esto lo podemos hacer con:

```
auto lo
iface lo inet loopback

# Interfaz Ethernet
auto eth0
iface eth0 inet static
        address         IP_INTERFAZ
        netmask         IP_MASCARA
        network         IP_RED
        broadcast       IP_Difusión
        gateway         IP_Router

# Interfaz Wifi
auto wlan0
iface eth0 inet static
        address         IP_INTERFAZ
        netmask         IP_MASCARA
        network         IP_RED
        broadcast       IP_Difusión
        gateway         IP_Router
        wireless_essid essid
```

- ***address*** : corresponde a la dirección IP de la tarjeta de red.
- ***netmask***: dirección IP con la que junto a address se identifica la dirección IP de la red.
- ***network*** : dirección que identifica a toda la red.
- ***broadcast*** : dirección que se utiliza para comunicarse con todos los equipos instalados en la red.
- ***gateway*** : ó puerta de enlace que identifica la interfaz del router/modem/server.
- ***wireless-essid*** (solo para WIFI) : nombre de la red inalámbrica que tengamos instalado.

Configuración dinámica

Si llegamos a no tener una ***ip estática***, si tenemos activado el servicio ***DHCP*** en nuestro ***router*** o si tenemos corriendo dicho servicio con un servidor, podemos simplificar la configuración de las interfaces con una configuración parecida a:

```
# The primary network interface
auto eth0
iface eth0 inet dhcp
```

Creación de interfaces virtuales

Se nos puede presentar el caso de que tengamos un ordenador portátil que lo conectemos a varias redes, con distintas configuraciones, ya sea en casa, en el trabajo, en la "zona wifi" del bar o de un "amigo". Para este caso podemos cambiar a mano la configuración del interfaz, o configurar **interfaces virtuales** en nuestro pc.

Necesitamos hacer una pequeña diferenciación entre interfaz física y lógica. La física está clara, pero podemos tener varias interfaces lógicas (que configuraremos) en una misma interfaz física. A la cual pertenecen; si se activa una interfaz lógica también se activará la física, no pudiéndose dar dos interfaces lógicas por el mismo interfaz físico.

Una manera de definir interfaces lógicas es identificando, en nuestro archivo de interfaces, la interfaz física con la lógica seguida de dos puntos (***:***), quedando algo tal que ***eth0:0***, donde ***0*** es el identificador de dicha interfaz. Quedando nuestro archivo interfaces, algo tal que:

```
# Interfaz Ethernet
iface eth0 inet static
        address         192.168.1.101
        netmask         255.255.255.0
        network         192.168.1.0
        broadcast       192.168.1.255
        gateway         192.168.1.1

iface eth0:0 inet dhcp
```

Donde vemos que ***eth0:0*** es la interfaz virtual. Para levantar dicha interfaz, debemos de introducir el comando :

$ ifup eth0:0

También podemos introducir directamente, en el archivo interfaces, el identificador de la interfaz lógica y su configuración; quedando algo como:

```
# Interfaz Ethernet
iface hogar inet static
        address         192.168.1.101
        netmask         255.255.255.0
        gateway         192.168.1.1

iface trabajo inet static
        address         X.X.X.X
        netmask         255.255.255.0
        gateway         X.X.X.X
```

Para conectarnos en este caso, debemos introducir el comando ***ifup eth0=hogar*** . Identificando tras el "igual" (=) la interfaz lógica que queremos activar.

Si queremos levantar/bajar todas las placas de red existe el script :

$ /etc/init.d/networking {start|stop}

Configuración de Bonding (bond)

Esto se utiliza para tener alta disponibilidad, etc.

¿ Qué es bond o bonding ?

El ***bond*** o ***bonding***, a grandes rasgos es un método de unión de interfaces, el cual nos permite desde balancear la carga, hasta soportar fallos de interfaces sin interrupción del servicio (alta disponibilidad).

Tipos de Bond.

Existen varios tipos o ***modos de bond***. Según nuestras necesidades, debemos optar por el que mejor se nos ajuste.

- ***Modo 0 (Roun Robin)***
  - *Transmite alternando interfaces, partiendo del primer esclavo.*
  - *Si hay balanceo de carga.*
  - *Si hay alta disponibilidad*

- ***Modo 1 (Active-Backup)***
  - *Usa solo un solo esclavo, solo en el caso de que falle pasa a usar el siguiente.*
  - *No hay balanceo de carga.*
  - *Si hay alta disponibilidad.*

- ***Modo 2 (Balance-XOR)***
  - *Se alterna el uso de uno u otro esclavo.*
  - *Si hay balanceo de carga.*
  - *Si hay alta disponibilidad.*

- ***Modo 3 (Broadcast)***
  - *Todo se manda por todos los esclavos.*
  - *No hay balanceo de carga.*
  - *Si hay alta disponibilidad.*

- ***Modo 4 (802.3ad)***
  - *Crea grupos que comparten la misma velocidad, pero las tarjetas deben soportar, IEEE 802.3ad.*
  - *Si hay balanceo de carga.*
  - *Si hay alta disponibilidad*

- ***Modo 5 (Balace-tbl)***
  - *Balancea todo el trafico de salida, y el trafico de entrada es recibido por el esclavo activo.*
  - *Si hay balanceo de carga.*
  - *Si hay alta disponibilidad.*

- ***Modo 6 (Balance-alb)***
  - *Igual que el anterior pero balancea también el trafico de entrada. El driver de las tarjetas debe soportar el cambio de MAC estando activas.*
  - *Si hay balanceo de carga.*
  - *Si hay alta disponibilidad.*

Configuración de Bond

Bajamos el paquete ***ifenslave-2.6.***

$ apt-get install ifenslave-2.6

1.- Creación de Alias para la carga de los modulos y elección del modo de Bond.

En el fichero:

$ cd /etc/modprobe.d

Definimos un alias para el ***bond***, así como seleccionamos el modo en el que correrá nuestro ***bond***.

$ vi aliases-bond.conf

alias bond0 bonding
options bonding mode=1 arp_interval=2000 arp_ip_target=192.168.0.200

options bonding mode=1

Donde "X" es el número del modo que queremos. (0..6)

2.- Cargamos el modulo de ***bonding***.

Para asegurarnos que el modulo esta cargado tras los cambios, ejecutamos:

$ modprobe bonding

3.- Configuramos las interfaces.

Por último queda definir las interfaces, tanto las de red (físicas), como la virtual del ***bond***.

$ vi /etc/network/interfaces

 # The loopback network interface
auto lo
iface lo inet loopback

 # The primary network interface
#allow-hotplug eth0
#iface eth0 inet dhcp

auto bond0
iface bond0 inet static
address 192.168.0.200
netmask 255.255.255.0
network 192.168.0.0
gateway 192.168.0.1
slaves ***eth0 eth1***
bond-mode active-backup
bond-miimon 100
bond-downdelay 200
bond-updelay 200

4.- Reinicio de los servicios de Red.

Para esto, basta con ejecutar como ***root***:

$ ifup bond0

5.- Ver el estado del ***bond***.

Una vez funcionando, podemos ver el estado del ***bond***, ejecutando el comando:

$ cat /sys/class/net/bond0/bonding/mode

Comando netstat

El comando ***netstat*** (network status) nos informa sobre la configuración y actividad de la red. Veamos las principales opciones de este comando:

- ***configuración de las interfaces de red*** : la opción ***-i*** nos mostrará la configuración de las interfaces de red activas y con la opción ***-e*** obtendremos información extendida (obtendremos la misma salida que con el comando **ifconfig**):

  $ netstat -ie

  ```
  Kernel Interface table
  eth0      Link encap:Ethernet  HWaddr 00:24:8c:7a:f3:d8
            inet addr:192.168.0.100  Bcast:192.168.0.255  Mask:255.255.255.0
            inet6 addr: fe80::224:8cff:fe7a:f3d8/64 Scope:Link
            UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
            RX packets:124083 errors:0 dropped:0 overruns:0 frame:0
            TX packets:51702 errors:0 dropped:0 overruns:0 carrier:31
            collisions:0 txqueuelen:1000
            RX bytes:47562122 (45.3 MiB)  TX bytes:7742430 (7.3 MiB)
            Interrupt:27

  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:555 errors:0 dropped:0 overruns:0 frame:0
            TX packets:555 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:0
            RX bytes:32128 (31.3 KiB)  TX bytes:32128 (31.3 KiB)
  ```

  Si añadimos la opción **-a** mostrará también las interfaces que no estén up.

- ***Enrutamiento***: la opción ***-r*** nos mostrará la tabla de enrutamiento (obtendremos la misma salida que con el comando *route*), para evitar el mecanismo de resolución de nombres y ver el enrutamiento con las direcciones IP añadiremos la opción:

  $ netstat -nr

  ```
  Kernel IP routing table
  Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
  ```

192.168.0.0 0.0.0.0 255.255.255.0 U 0 0 0 eth0
0.0.0.0 *192.168.0.1 0.0.0.0 UG 0 0 0 eth0*

- ***Conexione*s**: si queremos ver las conexiones ***tcp*** y ***udp***.

  $ netstat -apntu

```
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
PID/Program name
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      2060/sshd
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN      1706/cupsd
tcp        0      0 127.0.0.1:25            0.0.0.0:*               LISTEN      2006/exim4
tcp        0      0 192.168.0.100:22        192.168.0.101:59050     ESTABLISHED
21964/3
tcp6       0      0 :::80                   :::*                    LISTEN      1469/apache2
tcp6       0      0 :::22                   :::*                    LISTEN      2060/sshd
tcp6       0      0 ::1:631                 :::*                    LISTEN      1706/cupsd
tcp6       0      0 ::1:25                  :::*                    LISTEN      2006/exim4
tcp6       0      0 :::443                  :::*                    LISTEN      1469/apache2
```