Comandos para Administración de Redes

Comando	Descripción

arp *(Unix y MS Windows)*	**Muestra y modifica la memoria caché de ARP**
	El comando *arp* permite abrir y editar la memoria caché de la tabla ARP del sistema operativo, donde se indica la relación entre la MAC Address y la IP Address.
	`arp [OPCIONES] [HOST]`
	Con la opción *-a* se limita la salida a las entradas de un determinado nombre de host o de una dirección IP:
	`arp -a HOST`
	Si se ha de eliminar una entrada ya existente se utiliza *arp* con la opción *-d* (*delete*):
	`arp -d HOST`
	Para limpiar la cache ARP completamente:
	`ip -s -s neigh flush all`

dig *(Unix)*	**Solicita información del DNS**
	Herramienta de búsqueda con la cual pedir información a los servidores DNS.
	`dig [@SERVIDOR] [DOMINIO] [TIPO]`
	SERVIDOR es el servidor DNS al cual se realiza la petición y donde se encuentra la información solicitada. Si no se indica un servidor, *dig* indaga en el servidor DNS estándar que figura en el archivo `/etc/resolv.conf`.
	DOMINIO equivale al nombre del dominio sobre el cual se busca la información almacenada en el DNS.
	Y TIPO permite determinar el tipo de petición: ANY (cualquier entrada), A (record IPv4 de un host) o AAA (record IPv6 de un host). El tipo estándar es A.
	Utiliza la opción *-x* para preguntar por el nombre de dominio de una dirección IP dada en una búsqueda inversa:
	`dig [@SERVIDOR] [-x DIRECCIÓN IP]`
	En este caso no es necesario indicar nombre, tipo o clase.

ftp *(Unix y MS Windows)*	**Transfiere archivos por FTP** Permite intercambiar archivos entre el sistema local y otro equipo en la red. `ftp [OPCIONES] [HOST[PUERTO]]` El direccionamiento tiene lugar con un nombre de host o una dirección IP. El número de puerto es opcional. Durante el proceso de conexión, por lo general será necesario introducir un nombre de usuario y su contraseña. Cuando los datos son correctos, *ftp* inicia un intérprete de líneas de comandos que recibe las entradas del usuario en forma de comandos. Este programa soporta diversas órdenes con las cuales explorar y gestionar el sistema de archivos del equipo de destino, así como transferir archivos de un sistema a otro. Con *help* se obtiene listado.
ifconfig	**Actualmente deprecated** Permite ver la configuración de red de las tarjetas instaladas en el equipo, gestionar

(Unix)	interfaces, configurar las interfaces y todo lo relacionado con la propia red.
	`ifconfig [INTERFAZ] [OPCIONES]`
	Asignar una dirección IP:
	`ifconfig eth0 192.168.0.2 netmask 255.255.255.0`
	Habilitar una interfaz de red:
	`ifup eth0`
	Deshabilitar una interfaz de red:
	`ifdown eth0`
	Modificar el MTU:
	`ifconfig eth0 mtu XX`

ipconfig
(MS Windows)
	Muestra información de red
	Muestra los datos esenciales como la Dirección IP, la Máscara de red y la Puerta de enlace, para cada adaptador encontrado.
	Para mostrar toda la informacion de los adaptadores
	`ipconfig /all`
	Para liberar y renovar la dirección IP.
	`ipconfig /release`
	`ipconfig /renew`

ip *(Unix)*	**Administra interfaces de red** El programa *ip* forma parte del paquete de utilidades *iproute2*, con el cual se pueden consultar y configurar interfaces de red en el terminal. *`ip [OPCIONES] OBJETO [COMANDO [ARGUMENTOS]]`* El programa soporta objetos como *address* (dirección IP), *link* (interfaces de red), *route* (entrada en la tabla de enrutamiento) o *tunnel*, sobre el cual se pueden aplicar comandos de búsqueda como *add*, *change*, *del*, *list* o *show.* Si, por ejemplo, se quiere ver la dirección IP de una determinada interfaz de red (eth0): *`ip address show dev eth0`* Los objetos y las órdenes también se pueden enviar en forma abreviada: *`ip a s dev eth0`* Si lo que deseas es obtener toda la información sobre una interfaz de red (eht0) combina el comando *ip* con el objeto *link*, la orden *show* y el argumento *dev eth0*: *ip link show dev eth0* Para activar o desactivar una interfaz como eth0, procede así: *`ip link set eth0 up`* *`ip link set eth0 down`* Con su gran paleta de funciones, *iproute2* sustituye a una serie de herramientas de red más antiguas como *ifconfig*, *route* y *netstat*.
iw *(Unix)*	**Consulta y configura interfaces inalámbricas** El programa *iw* se utiliza para configurar interfaces inalámbricas y se ha consolidado como la alternativa más reciente a *iwconfig*. *`iw [OPCIONES] OBJETO [COMANDO]`* Algunos objetos que se pueden usar con *iw* son: dev NOMBRE_DE LA_INTERFAZ = Interfaz de red phy NOMBRE_DEL_DISPOSITIVO = Dispositivo inalámbrico (vía nombre) phy#ÍNDICE_DEL_DISPOSITIVO = Dispositivo inalámbrico (vía índice)

	reg = Agente regulatorio para configurar la región y el país
	Muestra las características de los dispositivos en todas las interfaces WI-FI:
	`iw list`
	Abre el estado de la conexión (tasa de transferencia y calidad de la señal) de una interfaz inalámbrica (wlan0):
	`iw dev wlan0 link`
	Escanea el entorno inalámbrico:
	`iw dev wlan0 scan`
	Mostrar la configuración regional:
	`iw reg get`
	Editar la configuración regional:
	`iw reg set DE`
	Solicitar las características de los dispositivos (wlan0):
	`iw list dev wlan0`
	Características de los dispositivos en detalle:
	`iw dev wlan0 station dump`
	Consultar eventos:
	`iw event`
	Las opciones *-f*, *-t* y *-r* entregan una salida ampliada con notificaciones de error sobre el estado de la conexión y la fecha.

lsof *(Unix)*	lista los puertos abiertos en el servidor
mtr *(Unix)*	**Verificar saltos de routers** MTR o My Traceroute nos permite ver los saltos de los routers y hacerles un ping a cada uno. Esto es muy útil para determinar cual de estos routers son los que tienen demoras en el trafico de red. `mtr [HOST]`

netstat
(Unix y MS Windows)
	Consulta el estado de las interfaces de red
	Sirve para consultar el estado de las interfaces de red.
	`netstat [OPCIONES]`
	Ver las conexiones activas:
	`netstat -a`
	Desplegar puertos de escucha:
	`netstat -l`
	Para ver los programas asociados con los sockets abiertos:
	`netstat -p`
	Ver el puerto en uso por un programa:
	`netstat -ap | grep (Programa)`
	Utilizado sin opción, el comando entrega todos los sockets abiertos en el terminal. También se pueden utilizar las siguientes opciones para consultar tablas de enrutamiento (*-r*), conexiones enmascaradas (*-M*) o mensajes de enlace de red (*-N*).
	Una alternativa a *netstat* es el programa *ss* incluido en el paquete de herramientas *iproute2*.

nmap *(Unix y MS Windows)*	NMAP se considera el padre de los escáneres de red generales. Aunque hoy en día existen herramientas más fiables para algunas tareas (p.e: Fping), NMAP no deja de

	ser una herramienta muy versátil para escanear redes. Sirve para determinar qué hosts están vivos en una red y para hacer escaneos de diversos modos.
nmcli *(Unix)*	**Muestra informacion sobre el NetworkManager** nmcli es una herramienta para controlar el NetworkManager y reportar el estado de la red. nmcli se utiliza para crear, visualizar, editar, borrar, activar y desactivar conexiones de red, así como para controlar y visualizar el estado de los dispositivos de red. `nmcli device show`

nslookup
(MS Windows)
	Consulta información en el DNS
	Sirve para resolver nombres. El comando está disponible en dos modos operativos: interactivo y no interactivo.
	El modo interactivo se inicia escribiendo en el terminal *nslookup* a secas.
	`nslookup`
	El programa está preparado para recibir órdenes. Si queremos consultar la dirección IP de un dominio, se introduce el nombre del host. Si queremos iniciar una consulta inversa, se introduce una dirección IP.
	El programa *nslookup* utiliza automáticamente el servidor DNS configurado en el sistema.
	Para finalizar *nslookup* se introduce en el terminal el comando *exit*.
	Si queremos iniciar *nslookup* en el modo no interactivo se invoca el programa en combinación con un nombre de host o una direccion IP.
	`nslookup [OPCIONES] [HOST/IP ]`
	En GNU LInux está obsoleto oficialmente, suele recomendarse a los usuarios utilizar *dig* en su lugar.

ping *(Unix y MS Windows)*	**Comprueba la conexión de red** (Packet INternet Groper) hace uso del protocolo ICMP para comunicarse con otros dispositivos y verificar su estado. `ping [OPCIONES] DESTINO` Para comprobar la conexión de red, *ping* envía un pequeño paquete de datos al sistema de destino indicado (dominio o IP) y evalúa el tiempo que transcurre hasta que se registra una respuesta. Además de registrar la franja de tiempo que pasa entre el envío del paquete de datos y la recepción de la respuesta (Round trip time o RTT), *ping* también escribe la dirección IP del sistema de destino en el terminal. Esto hace que este comando también se utilice para averiguar la dirección IP de un dominio. Si *ping* no se ve acompañado por ninguna opción, el programa se ejecuta hasta que se finaliza a mano con el atajo [CTRL] + [C] y envía al sistema de destino una petición *ping* por segundo. Pero si ya al comienzo se quiere definir un punto de finalización, se utilizan las opciones *-c NÚMERO* (número de peticiones *ping* que se han de enviar) o *-w SEGUNDOS* (franja temporal en segundos tras la cual *ping* finaliza automáticamente).

route
(Unix)
	Muestra y edita tablas de enrutamiento IP
	Con el comando *route* se pueden consultar y editar las tablas de IP routing del kernel.
	`route [OPCIONES]`
	`route [OPCIONES] [add|del] [-net|-host] OBJETIVO`
	Si utilizas el comando sin opción se obtiene una tabla de enrutamiento completa del

	núcleo:
	`route`
	Si quisieras añadir una ruta de enrutamiento a una red, utiliza la opción *add*:
	`route add -net 10.0.0.0`
	Si el destino consiste en una subred, se ha de proporcionar la máscara de subred con la opción *netmask MÁSCARA:*
	`route add -net 10.0.0.0 netmask 256.245.155.0`
	También se puede configurar una ruta a un PC:
	`route add -host 218.89.72.191`
	Si el sistema dispone de varias interfaces de red, se ha de indicar con la opción *dev* INTERFAZ cuál de ellas se tiene que utilizar:
	`route add -net 10.0.0.0 netmask 256.245.155.0 dev eth0`
	En caso que el destino solo se pueda alcanzar con un router, también se ha de indicar con la opción *gw ROUTER*.
	`route add -net 10.0.0.0 netmask 256.245.155.0 gw 10.0.1.261`
	Para borrar una ruta se utiliza *del*:
	`route del -host 218.89.72.191`

scp
(Unix)
	Transfiere archivos con SCP
	Copia datos de un equipo a otro utilizando también para ello el protocolo de red SSH.
	El programa cliente funciona casi como la opción *cp,*
	`scp [OPCIONES] ARCHIVO [[user@]remote_host:]RUTA`
	A la indicación de la ruta del equipo le preceden el nombre de usuario y el nombre del host remoto. Los archivos locales se pueden direccionar mediante rutas relativas o absolutas.
	`Scp /home/matias/images/image.jpg matias@ej.com:/home/matias/archiv`
	El archivo *image.jpg* se copia desde el directorio local *images* en el directorio *archiv* alojado en un equipo con la dirección ej.*com*.
	El programa *scp* también soporta la transferencia de datos a la inversa, así como entre dos sistemas remotos.
	`scp [OPCIONES] [[user@]host:]ARCHIVO RUTA`
	`scp [OPCIONES] [[user@]host1:]ARCHIVO [[user@]host2:]RUTA`
	Gracias a otras opciones se puede configurar el modo de la transferencia y el cifrado.

sftp *(Unix)*	**Transfiere archivos por SFTP** El programa *sftp* sirve para transferir datos en una red igual que *ftp*, aunque las operaciones tienen lugar con una conexión cifrada SSH (Secure shell). Como *ftp*, *sftp* también establece una conexión a un equipo de destino en la red y arranca a continuación un modo de comando interactivo.
ss *(Unix)*	**Comprobación del rendimiento de la conexión** El comando de estadísticas de socket ss es un reemplazo para netstat, es más rápido y brinda más información. `ss | less` Este comando genera todas las conexiones de socket TCP, UDP y UNIX y canaliza el resultado al comando less para una mejor visualización. Combinado con -t para mostrar los sockets TCP o -u para mostrar UDP o -x para mostrar los sockets de UNIX.

	Para enumerar todos los sockets TCP establecidos para IPV4, usa el siguiente comando: `ss -t4 state established`

tcpdump
(Unix)
	TCPDump es un comando avanzado utilizado para inspeccionar el tráfico de las diferentes interfaces de una máquina y así poder obtener los paquetes intercambiados.
	`tcpdump -i <network_device>`
	Se puede especificar un protocolo (TCP, UDP, ICMP y otros):
	`tcpdump -i <network_device> tcp`
	Además, puede especificar el puerto:
	`tcpdump -i <network_device> port 80`
	tcpdump seguirá ejecutándose hasta que se cancele la solicitud; es mejor usar la opción -c para capturar un número predeterminado de eventos como este:
	`tcpdump -c 20 -i <network_device>`
	También puedes especificar la IP a capturar utilizando la opción src o la dst.
	`tcpdump -c 20 -i <network_device> src XXX.XXX.XXX.XXX`
	Se puede volcar a fichero la salida para luego analizarla con otros sniffers más potentes y con interfaces gráficos como Wireshark. Para MS Windows, debe utilizarse WinDump.

telnet *(Unix y MS Windows)*	**Conexión entre equipos** Establece conexiones remotas con otras PC, servidores, y dispositivos con un sistema compatible en el acceso mediante este sistema de comunicación. De forma predeterminada se utiliza el puerto de conexión 23. `telnet <IP/dominio del host>` Para establecer una conexión entre dos equipos con *telnet*, se debe tener un cliente en el terminal que se desea conectar, y un servidor en la máquina a la que se desea acceder y el puerto 23 abierto en la máquina de destino. Se deberá abrir una sesión en la máquina de destino en la que exista una o varias cuentas de usuarios que tengan permitido el acceso. En definitiva, para acceder con un cliente a una máquina destino esta deberá contener una cuenta de usuario habilitada para el acceso, y para conectar se necesitara conocer tanto el nombre como la contraseña de usuario para establecer la comunicación.
traceroute *(Unix)* ***tracert*** *(MS Windows)*	**Sigue a los paquetes de datos** Utilizar el comando *traceroute* para trazar la ruta que sigue un paquete de datos IP entre un dispositivo y otro en la red. Indica los saltos entre routers. `traceroute [OPCIONES] HOST` Con *traceroute* se puede averiguar por qué router y por qué nodos ha pasado un paquete IP en su camino hacia el dispositivo de destino para, por ejemplo, investigar la causa de un desfase. Así también, nos muestra la latencia generada durante todo ese camino recorrido y la cantidad de pérdida de datos, si es que lo hubiera.
whois *(Unix y MS Windows)*	**Información sobre dominios** Brinda información detallada respecto al dominio consultado. Funciona como cliente para el protocolo del mismo nombre «whois» y provee información de recursos de red gracias a su gran base de datos. `whois [HOST]`