

Comandes de xarxa

Configuració bàsica de la xarxa:

Les tarjes de xarxa en sistemes Linux es poden anomenar de les següents maneres:

- lo : Correspon a la tarja "loopback". Recordem que hem dit que aquesta tarja físicament no existeix (és un "invent" del sistema operatiu) i sol tenir sempre té la IP 127.0.0.1/8. Serveix per establir una connexió amb sí mateixa, de tal forma que podem tenir a la mateixa màquina un programa client que connecti a un programa servidor sense sortir "a fora".
- eno1, eno2 : Tarjes Ethernet integrades a la placa base ("on-board")
- ens1, ens2 : Tarjes Ethernet PCI ("slot")
- enp2s0, p3p1....: Tarjes Ethernet que no es poden localitzar d'altra forma degut a limitacions de la BIOS
- wlp2s0,... : Tarjes WiFi

Dins de les màquines virtuals de VirtualBox les tarjes de xarxa agafen sempre un nom concret: la tarja corresponent a la primera pestanya del quadre de configuració de xarxa s'anomenarà "enp0s3" dins del sistema virtualitzat, la segona "enp0s8", la tercera "enp0s9" i la quarta "enp0s10".

Veure la configuració actual

Per veure l'estat i configuració de les tarjes detectades es pot fer servir la comanda **ip address show** (o bé **ip address show dev *nomTarja*** si només es vol obtenir la informació d'una tarja determinada). Concretament ens mostra:

- Les direccions MAC de les tarjes
- El seu estat respectiu (UP, DOWN)
- Les seves direccions IP respectives (i la màscara corresponent)
- Altres dades (com si permet l'enviament "broadcast", si està en mode "promiscu", etc).

NOTA: La comanda *ip address show* es pot escriure de forma més curta així: *ip a s* . Fins i tot, es pot deixar d'escriure el verb *show* (o *s*) perquè és l'acció perfecta (per tant, es pot fer *ip address* o *ip a* i seria el mateix). També és útil el paràmetre *-c* (així: *ip -c a s*) per veure les dades més rellevants en colors.

Per saber, en canvi, quina és la porta d'enllaç configurada a la nostra màquina, s'ha de fer servir la comanda **ip route show** (o bé **ip route show dev *nomTarja*** o les seves variants *ip route*, *ip r s* o *ip r*). Aquesta comanda ha de mostrar una línia que començarà amb l'expressió "default via" seguida de la direcció IP de precisament la porta d'enllaç establerta. Aquesta comanda pot mostrar més línies, però no ens interessaran gaire...(potser la més curiosa és una que serveix per indicar que no cal cap porta d'enllaç per comunicar-se amb les màquines que precisament pertanyin a la mateixa xarxa a la què pertany la nostra màquina).

Per saber la direcció IP del servidor DNS configurat a la nostra màquina (o les IPs...si n'hi ha més d'una es prova de connectar a la primera i si aquesta falla llavors es prova la segona, i així) es pot consultar l'arxiu **/etc/resolv.conf** (concretament, les línies que comencen per la paraula *nameserver*). Aquests servidors seran els que totes les aplicacions del sistema (des del ping fins el navegador) faran servir per esbrinar quina és la IP del nom que l'usuari hagi escrit. El contingut d'aquest arxiu sol ser gestionat per diferents programes (com ara pot ser el client *dhclient*,

l'aplicació NetworkManager, el servei "networking", el servei "systemd-networkd/resolved", etc) i és per això que no es recomana modificar-lo manualment ja que els canvis realitzats a mà podrien "matxacar-se" sense avisar en qualsevol moment per qualsevol d'aquests programes. En aquest sentit, aquests programes (dhclient, NetworkManager, "networking", "systemd-networkd/resolved", etc) guarden els servidors DNS que usen dins dels seus propis arxius de configuració i els manipulen allà de forma autònoma (per exemple NetworkManager usa /var/run/NetworkManager/resolv.conf, "systemd-resolved" usa /run/systemd/resolve/resolv.conf, "networking" usa la línia dns-nameservers dins de /etc/network/interfaces, etc) però a més sempre vinculen en forma d'enllaç simbòlic el seu arxiu propi respecte a l'arxiu comú /etc/resolv.conf per a què els programes que facin servir aquest arxiu comú no tinguin problemes en trobar els servidors DNS. Als exercicis es veuran casos pràctics de com gestionar tot això.

Establir una configuració estàtica de forma temporal

*Per assignar una IP/màscara concreta a una tarja: **ip address add v.x.y.z/n dev nomTarja**

*Per esborrar una IP/màscara concreta a una tarja: **ip address del v.x.y.z/n dev nomTarja**

NOTA: També es pot fer **ip [-4|-6] address flush dev nomTarja** si el que es vol és esborrar de cop qualsevol de les eventuais diferents direccions IP que pugui tenir la tarja indicada

*Per assignar la porta d'enllaç concreta a una tarja: **ip route add default via v.x.y.z dev nomTarja** . Abans, però, s'hauria d'esborrar la que hi havia assignada abans (si no es fa dóna error), així: **ip route del default dev nomTarja** .

NOTA: També es pot escriure **ip route add 0.0.0.0/0 via v.x.y.z dev nomTarja**. És equivalent.

NOTA: De forma alternativa, en comptes de fer *ip route del ...* i després *ip route add ...*, el canvi de porta d'enllaç per defecte es podria fer directament en un sol pas, així: **ip route change default via v.x.y.z dev nomTarja**

NOTA: També es pot indicar que es vol fer servir una determinada porta d'enllaç només per arribar a una xarxa-destí concreta. En aquest cas, llavors, no estariem parlant de porta d'enllaç "per defecte" sinó d'una porta d'enllaç "específica". La porta d'enllaç "per defecte" seria usada llavors que el sistema hagués comprovat que el destí desitjat no forma part del conjunt de destins indicats a portes d'enllaç específiques. Per crear una porta d'enllaç específica cal executar la comanda **ip route add ip.Xarxa.Desti/Mascara via v.x.y.z dev nomTarja**. Es pot afegir a més un darrer paràmetre **metric n°**, que indica la preferència de la ruta en el cas de què hi haguessin varies que portessin al mateix destí (a mode de "backup"): un n° menor indica una major preferència. **NOTA:** Un cop assignada una direcció IP a una tarja, el sistema calcula automàticament la seva direcció IP de xarxa corresponent i genera una ruta a ella (és per això que és necessari indicar la màscara en *ip address add...*) Per exemple, si s'assigna la IP 203.0.113.25/24 a la tarja enp0s3, es crearà automàticament una ruta a la xarxa 203.0.113.0/24 directa, de manera que el sistema sabrà que per comunicar-se amb hosts d'aquesta xarxa no necessitarà cap porta d'enllaç intermediària sinó que ho podrà fer directament.

*Per activar/desactivar una tarja: **ip link set {up|down} dev nomTarja**

Establir una configuració estàtica de forma permanent (en sistemes Debian clàssics)

Totes les comandes anteriors, no obstant, només "funcionen" mentre la màquina es manté encesa: si s'apaga llavors les direccions IP/màscares i portes d'enllaç configurades amb les comandes "ip" anteriors es perden i cal, doncs, tornar-les a executar un altre cop al següent inici. Per tal de què la configuració desitjada de IP/màscara i porta d'enllaç (i servidor DNS també, gestionat amb algun dels programes comentats en paràgrafs anteriors) per una determinada tarja de xarxa es mantingui de forma permanent a cada reinici de la màquina, cal escriure els valors adients en un determinat arxiu. En sistemes Debian/Ubuntu, aquest arxiu s'anomena /etc/network/interfaces

i ha de tenir un aspecte similar al següent (les línies que comencen per # són comentaris; les tabulacions són opcionals):

```
#Les línies "auto" serveixen per activar la tarja en qüestió (en aquest cas, la tarja "lo")
auto lo
#La línia següent indica que la tarja "lo" és de tipus "loopback" (i que, per tant, tindrà la IP 127.0.0.1)
iface lo inet loopback
#En el mateix arxiu es poden configurar totes les tarjes que es vulguin: la següent s'anomena enp3s0
auto enp3s0
#La paraula "static" indica que els valors d'IP, màscara, etc són fixes a cada reinici
iface enp3s0 inet static
#A continuació s'indiquen els valors d'IP, màscara, porta d'enllaç i servidors DNS que es volen assignar
    address v.x.y.z
    netmask w.w.w.w
    gateway v.x.y.z
    dns-nameservers v.x.y.z v.x.y.z
```

NOTA: Atenció, la línia "dns-nameservers" de l'arxiu anterior només funciona (és a dir, es copien els servidors DNS indicats allà a l'arxiu central del sistema on han de ser-hi per tal de ser utilitzats: /etc/resolv.conf) si hi ha instal·lat un paquet anomenat "resolvconf". Si no ho està, aquestes línies no es tindran en compte

Aquest arxiu és llegit per un servei del sistema (un dimoni) que es posa en marxa automàticament en arrencar la màquina i que s'anomena "networking". Això vol dir que en qualsevol moment que fem un canvi dins d'aquest arxiu, per a què es tingui en compte o bé caldrà reiniciar la màquina o bé simplement reiniciar el servei, així: `sudo systemctl restart networking`

Establir una configuració estàtica de forma permanent (en sistemes Systemd)

"Systemd-networkd" és un dimoni que gestiona les configuracions de les diferents interfícies de xarxa (físiques i/o virtuals) d'un sistema Systemd, representant, doncs, una alternativa al dimoni "networking" de sistemes Debian així com també al scripts `ifcfg-*` clàssics de Fedora/Suse o al Network Manager integrat a molts escriptoris.

Per començar a utilitzar aquest dimoni és recomanable aturar primer la "competència" (per exemple, en el cas d'Ubuntu executant `sudo systemctl disable networking && sudo systemctl stop networking`) i llavors encendre'l juntament amb el servei "Systemd-resolved" (així, per exemple: `sudo systemctl enable systemd-networkd && sudo systemctl start systemd-networkd && sudo systemctl enable systemd-resolved && sudo systemctl start systemd-resolved`).

NOTA: A l'igual que passava amb el paquet "resolvconf" a sistemes Debian, es necessita tenir un servei addicional instal·lat (i funcionant) al sistema anomenat "systemd-resolved" si es volen especificar entrades DNS explícites en els arxius `.network` (o bé si s'obtenen via DHCP). Aquest servei el que fa és, a partir d'aquestes entrades, modificar l'arxiu `/run/systemd/resolve/resolv.conf`, el qual, per compatibilitat amb molts programes tradicionals, hauria d'apuntar en forma d'enllaç suau a `/etc/resolv.conf` (`ln -s /run/systemd/resolve/resolv.conf /etc/resolv.conf`). Alternativament, es pot no fer servir systemd-resolved i modificar llavors l'arxiu `/etc/resolv.conf` a mà.

Es poden veure el nom (i tipus i estat) de les interfícies de xarxa actualment reconegudes al sistema (i el seu tipus i estat) mitjançant la comanda **networkctl list**. Si a la columna **SETUP** apareix "unmanaged" vol dir que aquesta interfície concreta no és gestionada per systemd-networkd sinó per algun altre servei alternatiu. Una altra comanda que dóna més informació és **networkctl status**. En qualsevol cas, per fer que se gestioni per systemd-networkd, per cada interfície cal crear un arxiu `*.network` dins de la carpeta `/etc/systemd/network` (i reiniciar el servei). En el cas concret

de voler assignar una IP estàtica, caldria, doncs, tenir un arxiu com aquest (anomenat per exemple `" /etc/systemd/network/lalala.network"`):

```
[Match]
  Name=enp1s0      #Identifica la tarja a la qual se li aplicarà la configuració
[Network]
  DHCP=no
  Address=10.1.10.9/24
  Gateway=10.1.10.1
  DNS=10.1.10.2    #Opcional (cal tenir el servei systemd-resolved funcionant)
  DNS=10.1.10.3    #Cada servidor DNS ha d'indicar-se en una línia separada
```

Els arxius de configuració de Systemd-networkd proporcionats per la distribució es troben a `/usr/lib/systemd/network` i els administrats per nosaltres s'han d'ubicar a `/etc/systemd/network`. Tots aquests arxius es llegeixen -sense distinció d'on estiguin ubicats- en ordre alfanumèric segons el nom que tenen, guanyant sempre la primera configuració trobada en cas de què afectés a la mateixa tarja. Això sí, si a les dues carpetes es troba un fitxer amb el mateix nom, el que hi ha sota `/etc/systemd/network` anul·la sempre al que hi ha sota `/usr/lib/systemd/network` (una conseqüència d'això és que si l'arxiu a `/etc/...` apunta a `/dev/null`, el que s'estarà fent és deshabilitar

Existeixen tres tipus diferents d'arxius de configuració: els `".network"` apliquen la configuració descrita sota la seva secció `[Network]` a aquelles tarjes de xarxa que tinguin una característica que concordi amb tots els valors indicats a les diferents línies sota la secció `[Match]` (normalment aquí només s'indica el seu nom mitjançant una única línia `"Name="`); els `".netdev"` serveixen per crear noves interfícies de xarxa de tipus virtual (`"bridges"`, `"bonds"`, etc) -la configuració de xarxa dels quals se seguirà indicant en el seu corresponent arxiu `.network`) i els `".link"` serveixen per definir noms alternatius a les tarjes de xarxa en el moment de ser reconegudes pel sistema (via `systemd-udev`).

A les línies sota la secció `[Match]` -per exemple, a `"Name="`, es pot utilitzar el comodí `*`. En aquesta línia en concret també es pot escriure un conjunt de noms separats per un espai en blanc a mode de diferents alternatives.

Als arxius `.network` hi pot haver una secció (no vista als exemples anteriors) titulada `[Link]` sota la qual poden haver varies línies més relacionades amb el comportament "hardware" de la tarja, com ara la línia `"MACAddress=xx:xx:xx:xx:xx:xx"`, la qual serveix per assignar a la tarja en qüestió una direcció MAC fictícia, la línia `"MTUBytes=n"`, la qual serveix per indicar el tamany de la MTU admesa (útil per exemple per activar els "jumbo frames" si es posa 9000 com a valor), o la línia `"ARP=no"` per desactivar el protocol ARP a la tarja en qüestió (activat per defecte).

Els arxius `.netdev` solen tenir només una secció titulada `[NetDev]`, la qual ha d'incloure dues línies obligatòriament: `"Name="` (per assignar un nom a la interfície virtual que es crearà) i `"Kind="` (per especificar el tipus d'interfície que serà: `"bridge"`, `"bond"`, `"vlan"`, `"veth"`, etc). En el cas de que sigui de tipus `"vlan"`, apareixerà llavors una secció titulada `[VLAN]` incloent com a mínim la línia `"Id="` per indicar el número d'VLAN que s'està creant.

Els arxius `.link` solen tenir una secció `[Match]` amb la línia `"MACAddress="` per identificar la tarja de xarxa en qüestió i una secció `[Link]` que serveix per manipular les característiques d'aquesta tarja, com per exemple el seu nom (amb la línia `"Name="` i, opcionalment, la línia `"Description="`). Si no es crea manualment cap fitxer `.link`, la majoria de distribucions ofereixen un arxiu `.link` per defecte, generalment anomenat `99-default.link` (i ubicat a `/usr/lib/systemd/network`); és per això que cal assegurar-se que els fitxers `.link "manuals"` tinguin un nom que asseguri la seva lectura abans de la de l'arxiu `99-default.link`.

Per més informació sobre les possibilitats que ofereixen tots aquests arxius, consultar les pàgines del manual `"systemd.network"`, `"systemd.netdev"` i `"systemd.link"`.

NOTA: No només existeixen el serveix `"networking"` i `"systemd-networkd"` per gestionar les tarjes de xarxa del nostre sistema. També podem trobar el servei `"NetworkManager"` (sobre tot a sistemes amb escriptori) i a les darreres versions d'Ubuntu el servei `"Netplan"`, entre d'altres. Als exercicis els estudiarem.

Establir una configuració dinàmica de forma temporal

En les configuracions anteriors, tant la temporal com la permanent, s'estableix una direcció IP/màscara + porta d'enllaç concreta, decidida per nosaltres. Aquest mètode pot ser útil per poques màquines, però en una xarxa amb moltes d'elles pot arribar a ser força farragós, a més de que fàcilment es poden cometre errors (IPs duplicades, IP no assignades). Un altre mètode per establir aquestes dades és el mètode "dinàmic", en el qual la màquina en qüestió no té assignada de forma fixa cap IP/màscara + porta d'enllaç + servidor DNS sinó que aquestes dades les pregunta a la

xarxa: allà hi haurà d'haver escoltant un ordinador executant un software especial anomenat "servidor DHCP", el qual serveix precisament per atendre aquestes peticions de "dades de xarxa" i assignar-les a qui les demani. D'aquesta manera, es té una gestió centralitzada del repartiment de direccions IP/màscara + porta d'enllaç + servidors DNS sense que calgui realitzar cap configuració específica a les màquines clients. Això sí, és clar: primer caldrà haver instal·lat i configurat convenientment a la nostra xarxa aquest software "servidor DHCP" (un exemple és el paquet "isc-dhcp-server"), tasca que no veurem (presuposarem que això ja està fet).

NOTA: Segons com es configuri el servidor DHCP, aquest podrà assignar més o menys quantitat d'IPs, assignarà sempre la mateixa IP a la mateixa màquina (identificada per la MAC de la seva tarja), etc...però això ja està fora del nostre control si no som l'administrador de la xarxa.

Per demanar en un moment determinat una configuració completa de xarxa (IP, màscara, porta d'enllaç, servidors DNS, etc) a un servidor DHCP ja funcionant a la nostra LAN hem d'executar un client DHCP, com ara la comanda *dhclient* :

dhclient nomTarja Demana -per a la tarja indicada- les dades de xarxa (IP, màscara, porta d'enllaç, servidor DNS, etc) a algun servidor DHCP que estigui escoltant a la LAN de la nostra màquina

- v Mostra per pantalla tot el procés de petició i resposta (útil per veure si va)
- r Esborra totes les dades de xarxa que pugui tenir actualment la tarja indicada

Establir una configuració dinàmica de forma permanent (en sistemes Debian clàssics)

Per demanar les dades de xarxa a algun servidor DHCP de la nostra xarxa, en comptes d'haver de fer-ho manualment amb la comanda *dhclient*, es pot fer de forma automàtica cada cop que la nostra màquina arrenqui (de manera que nosaltres no haguem de fer res i ja tinguem, si tot va bé, aquestes dades ja assignades un cop iniciem sessió). Això s'aconsegueix simplement escrivint les següents línies a l'arxiu */etc/network/interfaces* :

```
auto enp3s0
iface enp3s0 inet dhcp
```

Establir una configuració dinàmica de forma permanent (en sistemes Systemd)

En el cas concret de voler assignar una IP dinàmica en un sistema amb el dimoni *systemd-networkd* funcionant, caldrà tenir un arxiu com aquest (anomenat per exemple *"/etc/systemd/network/lalala.network"*):

```
[Match]
Name=enp1s0
[Network]
DHCP=yes #També podria valer ipv4 o ipv6 segons el tipus de direccions IP que volem rebre
```

Comandes bàsiques de xarxa:

ping ip.oNom.Maq.Remota Comprova si la màquina remota indicada respon. Serveix, per tant, per saber si hi ha connexió de xarxa amb aquella màquina (si no, podria ser degut a qualsevol causa: cable mal endollat o trencat, màquina remota apagada, etc). En aquest sentit, són interessants les dades estadístiques que apareixen al final (paquets enviats, rebuts, perduts, etc) i el temps que han trigat en enviar-se aquests paquets de prova (i rebre's la resposta) -i així comprovar la saturació del medi.

-n No resol noms (és a dir, no fa la consulta prèvia al servidor DNS del sistema). Per tant, només fa que funcioni indicant direccions IP

-c n^o Número de paquets de prova que s'enviaran (si no s'indica, són infinits i cal aturar l'enviament pulsant CTRL+C)

-i n^o Número de segons que s'espera per enviar el següent paquet

-f Mode "flood". Envia paquets a la màxima velocitat possible, mostrant un punt per cada paquet enviat i esborrant-lo per cada resposta rebuda: per tant, per anar bé caldria que només es veiés un punt i anés desapareixent: si es veuen molts punts és que hi ha pèrdua de paquets. Cal ser root per a què funcioni

-I eno1 Indica la tarja de xarxa per la qual s'enviaran els paquets (per si la màquina tingués més d'una)

mtr ip.oNom.Maq.Remota Serveix per conèixer el camí seguit per un paquet des de la màquina origen fins l'indicada, mostrant la IP (o nom) de tots els routers intermitjos a través dels quals va passant. També mostra estadístiques dels temps emprats en cada paquet, el millor temps, el pitjor, els paquets perduts, etc

-n No resol noms (és a dir, no fa la consulta prèvia al servidor DNS del sistema).

-c n^o Número de paquets de prova que s'enviaran (si no s'indica, són infinits i cal aturar l'enviament pulsant CTRL+C)

-i n^o Número de segons que s'espera per enviar el següent paquet

ss Mostra dades sobre les connexions existents (o que poden existir) a la nostra màquina. Concretament, mostra l'estat de la connexió (els més habituals són ESTABLISHED i LISTEN -aquest últim indica que el port està obert però sense connexió-... altres estats sovint són temporals i acaben derivant en una connexió establerta o bé desapareixent), mostra la IP i el port local utilitzats per establir la connexió (o per escoltar, segons) i la IP i port remot on la corresponent IP+port local estan connectats .

NOTA: Les columnes "Recv-Q" i "Send-Q" mostren la quantitat de bytes que estan actualment al buffer temporal de memòria que gestiona el kernel per tal de regular la recepció i enviament de dades, respectivament. El normal és que valguin 0, indicant així que no hi ha cap dada d'entrada que s'estigui esperant a ser processada pel sistema ni cap dada de sortida que s'estigui esperant a efectivament sortir, respectivament

-t Mostra només les connexions TCP actuals

-u Mostra només les connexions UDP actuals

-n No resol noms (és a dir: mostra IPs i ports en format numèric en comptes de amb noms)

-a (Combinat amb -t i/o -u): Mostra, a més de les connexions actuals, els ports a l'escolta

- l (Combinat amb -t i/o -u): Mostra només els ports a l'escolta (les connexions actuals no)
- p (Combinat amb -t i/o -u): Mostra 1 columna més: l'executable "al darrera" de cada port local
- s Mostra un resum amb estadístiques

ncat ip.oNom.Maq.Remota n°port Client Netcat que ve dins del paquet "nmap": realitza una connexió (TCP) a la màquina i port indicat. Es pot afegir el paràmetre -v (mode verbós) i -n (no resol noms), entre altres.

-v Mode verbós (-vv és més verbós i -vvv més encara)

ncat -l -k -p n° Servidor Netcat: posa a l'escolta el port (TCP) indicat. El paràmetre -l serveix per "obrir" el port, el paràmetre -p serveix per indicar el número de port a obrir i el paràmetre -k permet que s'hi puguin connectar més d'un client a la vegada.

-e /ruta/comanda Tot el que es rebí de la xarxa serà passat a la comanda indicada, la sortida de la qual serà retornada al client. Si la comanda indicada fos /bin/bash, l'entrada s'entendrà com una comanda a executar (i la sortida serà la sortida de la comanda executada).

Exemples Ncat

***Chat**

Servidor: *ncat -l -p 5588 <--->* Client: *ncat ipServidor 5588*

El servidor se pone a escuchar en el puerto 5588 (por defecto siempre es TCP), con lo que todo lo que le llegue de la red -es decir, del cliente-lo pasará a la stdout (pantalla), y todo lo que escriba por stdin (teclado) pasará a la red -es decir, hacia el cliente-. Lo mismo ocurre en el otro lado de la comunicación. Si se añade el parámetro -k al servidor, múltiples clientes podrán enviar mensajes al servidor y este, lo que envíe, lo enviará a todos sin discriminación

***Envío de un archivo**

Servidor: *ncat -l -p 5555 < archivo <--->* Client: *ncat ipServidor 5555 > archivo*

Muy similar a lo anterior: el servidor se pone a escuchar en el puerto 5555, pero en vez de responder por teclado a la stdin, la entrada proviene de un archivo, el cual esperará latente a que cuando se establezca una comunicación por ese puerto, su contenido viaje bit a bit por la red hacia el cliente, el cual lo recibirá y lo guardará en forma de archivo otra vez. Lo malo es que tal como se ha hecho, no se sabe cuándo se ha acabado la transferencia: hay que esperar un tiempo prudencial y entonces hacer Ctrl+C.

***Reproducción de audio en streaming**

Servidor: *ncat -l -p 5858 <archivo.mp3 <--->* Client: *ncat ipServidor 5858 | mpg123 -*

El ejemplo es idéntico al anterior, teniendo un archivo en este caso de audio. La única diferencia es que en el cliente, el archivo no se redirecciona para grabarlo en disco sino que se entuba a un reproductor de audio por consola, como mpg123 (el guión del final es para indicarle que el fichero o lista de reproducción le proviene de la tubería).

***Clonación de discos por red**

Servidor: *ncat -l -p 5678 | dd of=a.iso.gz <--->* Client: *dd if=/dev/sda | gzip -c | ncat ipServidor 5678*

El ejemplo es parecido al anterior: primero en el cliente se comprime bit a bit el contenido del disco "sda" y se le envía ya comprimido al servidor, el cual recibe este contenido binario y lo almacena en un archivo, bit a bit too.

nmap -sn { ipInici-ipFinal [altraIP ...] | ipAmbAsterisks }

Mostra quins ordinadors estan presents a la xarxa. Existeixen molts altres paràmetres d'escaneig (-sU, -sX, -sF, etc) que fan servir diferents tècniques més o - ràpides/sigiloses/precises, però no les veurem.

- v Mode verbós (-vv és més verbós i -bb més encara)
- n No resol noms

nmap -p n^o,n^o-n^o ipOrdinador Mostra quins ports (del rang indicat) té oberts un ordinador concret. Aquí també es poden fer servir diferents tècniques però tampoc aprofundirem

-O Mostra el sistema operatiu de l'ordinador i els programes "al darrera" dels ports oberts. Es pot combinar amb el paràmetre -sV, el qual mostra també les versions. El paràmetre -A és la combinació dels dos.

nslookup nomDNS [ip.serv.DNS] Client DNS que pregunta al servidor indicat o, si no s'indica cap, al que estigui configurat a /etc/resolv.conf. Normalment, a més de retornar la IP (o IPs equivalents) associades al nom indicat, també mostra els "àlies" que té aquest nom

NOTA: Un altre client similar és la comanda *host nomDNS [ip.serv.DNS]*

NOTA: Un altre client més complet és la comanda *dig [@ip.serv.DNS] nomDNS*. O *drill*

NOTA: Un altre client però només compatible amb *systemd-resolved* és *systemd-resolve*

whois dominiDNS Consulta els servidors WHOIS administrats per la IANA per esbrinar les dades administratives de l'empresa que ha llogat el domini indicat

wget https://url/dun/fitxer Descarrega al disc dur el fitxer indicat

-c Continua la descàrrega (si anteriorment va fallar) des d'allà on es va interrompre

-O *nom* Indica el nom que tindrà el fitxer un cop descarregat

-r Realitza una descàrrega recursiva si la URL indicada és la d'una carpeta en comptes de la d'un fitxer. Combinat amb el paràmetre -l *n^o* serveix per indicar fins a quin nivell (1=una subcarpeta, 2=dues subcarpetes) es vol descarregar...si no s'indica s'entén "infinit"

-N Descarrega només els arxius més nous que els locals

-A "*ext1*", "*ext2*", ... Descarrega només els arxius que trobi amb l'extensió indicada

El paràmetre contrari (descarrega tot excepte els arxius indicats) és -R

--no-parent No descarrega contingut anterior a la URL indicada

-nd Tot o descarrega a la mateixa carpeta local (sense respectar, doncs, la jerarquia de carpetes del lloc remot)

-k Un cop feta la descàrrega, transforma els enllaços per tal de què tot el contingut es pugui visitar offline (canvia les rutes absolutes per relatives i als recursos no descarregats els referencia amb la URL completa)

curl https://url/dun/fitxer Descarrega el fitxer indicat i mostra a pantalla el seu contingut

-o *nom* Descarrega el fitxer indicat i el guarda al disc dur amb el nom que s'especifiqui

-O Descarrega el fitxer indicat i el guarda al disc dur amb el nom que tingui l'original

-C - Continua la descàrrega des del n^o de byte indicat (si és un guió, serà a partir d'on es va parar la descàrrega -fallida- anterior del mateix fitxer)

-s Mode "silenciós" (no mostra ni les estadístiques de descàrrega ni els errors, res)

-sS Mode "silenciós" però mostrant els missatges d'error

-v Mode "verbós". Serveix per mostrar les capçaleres de client enviades a la petició

-I No descarrega el fitxer: només mostra la capçalera de resposta HTTP del servidor

-i Mostra a pantalla tant les capçaleres de resposta com el contingut del fitxer demanat

-D *nom* Guarda al disc dur, en forma de fitxer amb el nom indica, les capçaleres de resposta

-L Si el servidor web retorna un codi de redirecció (3xx), el segueix automàticament

-H "*capçalera:valor*" Realitza una petició indicant un valor concret per la capçalera HTTP de client indicada. Es poden posar múltiples paràmetres -H.

-X *tipo* Realitza una petició del tipus indicat (POST, PUT, etc). Per defecte són GET