

# Comandos de Red

- [ethtool](#)
- [Comando ip](#)
- [Comando ping](#)
- [dig](#)
- [Whois](#)
- [mtr](#)
- [ss](#)
- [nc / netcat](#)
- [curl](#)
- [vnstat y vnstat](#)
- [iperf](#)
- [nload / tcptrack / nethogs / tmux](#)
- [nmap](#)
- [nmcli](#)
- [ssh](#)
- [rsync](#)
- [Iptables](#)

# ethtool

Este comando se utiliza para obtener información sobre la o las tarjetas de red que puede tener nuestro sistema.

```
ethtool enp0s3
```

## A tener en cuenta

- Auto-negotiation. No es mala idea activar la opción auto-negotiation, ya que es procedimiento Ethernet por el cuál dos dispositivos se ponen de acuerdo para utilizar al mismo momento configuraciones compatibles y comunes entre ellos. En redes del tipo 1000BaseT la auto-negociación no se puede desactivar.
- Port: Twisted Pair. Cuando no indique que el puerto es Twisted Pair, no está diciendo que la conexión sale por un puerto RJ-45.
- Link detected. Nos está indicando que el cable está conectado.
- Auto MDI-X. Es el protocolo que se encarga que mediante hardware hace que sea innecesario crear un cable crossover para conectar dos equipos. Esta tecnología se encarga de hacer el cruzamiento.
- Wake-on. Es un estándar que nos permite encender remotamente computadoras apagadas. Para activar wake-on-lan:
  - El primer paso es configurar la BIOS para que permite que dispositivos PCI puedan despertar al equipo.
  - El comando para establecer el wake-on-lan en la tarjeta sería:

```
ethtool -s enp0s3 wol -g. (-g para que sea del tipo magicpacket).
```

## Otras maneras

Otra forma de obtener información sobre la tarjeta de red podría ser:

- **lspci | grep "Ethernet Controller"**
- **lshw -class network**. Si queremos obtener más información aún.

# Comando ip

Primero vamos a cambiar el nombre de nuestra tarjeta de red. Para ello creamos el archivo **80-net-setup-link.rules** dentro de la carpeta **/etc/udev/rules.d/** con el siguiente contenido.

```
SUBSYSTEM=="net",ACTION=="add",ATTR{address}=="Mac_de_la_tarjeta de red",NAME="NombreChulo"
```

Una vez reiniciemos podemos apreciar el cambio de nombre, si queremos que vuelva a tener el nombre anterior solo deberemos borrar el archivo creado en **/etc/udev/rules.d**.

## Ver Número MAC

```
ip link show
```

## Asignar IP

```
ip addr add 192.166.10.1 dev enp0s3
```

## Asignar nombre local a la máquina

```
hostname set-hostname Nombre_nuevo
```

## Desactivar y activar tarjeta de red

- Desactivar. **ip link set enp0s3 down**
- Activar.. **ip link set enp0s3 up**

## Activar modo promiscuo en la tarjeta

```
ip addr set promisc on dev enp0s3
```

## Cambiar nº mac de la tarjeta de red

```
ip addr set address 00:11:22:33:44:55 dev enp0s3
```

# Averiguar puerta de enlace

```
ip route
```

# Borrar Puerta de enlace

```
ip route del default via ip_puerta_enlace
```

# Activar nueva puerta de enlace

```
ip route add default via ip_nueva_puerta dev enp0s3
```

# Contenido tabla ARP

```
ip neigh
```

# Borrar contenido tabla ARP

```
ip neigh del dirección_ip dev enp0s3
```

# Agregar una entrada en la tabla ARP y que sea permanente

```
op neigh add dirección_ip lladdr N°_MAC dev enp0s3 nud perm
```

# Comando ping

Posibles opciones:

- -n. Solo admite direcciones ip.
- -c. Mandamos el número de paquetes que queramos.
- -i. Podemos marcar el intervalo de tiempo en el que se mantendrán activos los paquetes ICMP.
- -w. Marcamos el tiempo de espera de una respuesta, se lo excedé la descartará.
- -I. Elegimos la interfaz de red que queremos usar.
- -f. Fuerza Bruta de ping. En 20 seg es capaz de mandar 1935 paquetes. Es necesario ser root para poder ejecutarlo.
- 0. Podemos comprobar si nuestra tarjeta esta funcionando.

## resolv.conf

Ya sabemos que los servidores DNS que un sistema utiliza vienen listados en el archivo `/etc/resolv.conf`. De todas formas este fichero no suele editarse manualmente ya que es gestionado por diferentes aplicaciones o programas al mismo tiempo. Todos estos programas no modifican tampoco el fichero `/etc/resolv.conf` sino que lo hacen de forma coordinada a través del demonio `resolv`. Además este demonio tiene sus propios ficheros de configuración que podemos utilizar para gestionar el fichero `resolv.conf`:

- `/etc/resolvconf/resolv.conf.d/base`
- `/etc/resolvconf/resolv.conf.d/head`
- `/etc/resolvconf/resolv.conf.d/tail`

## getent hosts

Nos sirve para ver todas las máquinas que fueron asignadas en `/etc/hosts`

## dnsmasq

Al instalar `dnsmasq`, lo realmente estamos haciendo es activarlo, de tal manera que cualquier dirección ip o nombre

- Archivo de configuración. **`/etc/dnsmasq.conf`**.

- Opción no-resolv: Si no quieres que dnsmasq lea el archivo /etc/resolv o otro resolv
- Opción server: Nos permite añadir un nombre al servidor dhcp-dns. Solo para dominios no-públicos.
- Opción no-hosts: Si no queremos que se lea el archivo /etc/hosts
- Opción addn-hosts: Todo lo contrario que el anterior, podemos marcar un archivo diferente de /etc/hosts para que lea los hosts de la red.
- Opción domain-needed.
- Opción bogus-priv.
- Archivo de equipos conectados al servicio. **/var/lib/misc/dnsmasq.leases.**

# dig

## Información sobre una web

```
dig www.hola.co
```

## Consulta inversa

```
dig ip_hola._.com
```

## Question Section

Tiene que ver con la pregunta que le hacemos al servidor, es decir, es este caso hemos preguntado por la Opción

## Authority

Nos esta diciendo que la respuesta no la obtiene de sus propios datos almacenados, sino que ha tenido que pregun

## Encontrar ipv6

```
dig www.rebellion.org AAAA
```

## Encontrar alias

```
dig www.rebellion.org CNAME
```

## Opciones dig

- no-course: Marcamos si queremos o no consultas recursivas
- trace: Nos permite ver que dns hay por encima del que estamos consultado. Nos permite marcar una ruta hacia los DNS superiores.
- short (seria una combinació de +nostats, +nocomments i algun més). Da una respuesta concisa, el modo por defecto es "verbose" que esta más detallado.

- tries=4. Nos sirve para establecer el número de intentos que hace la consulta UDP hacía el servidor. Por defecto son 3 intentos.
- time=3. Nos sirve para establecer el tiempo que estará intentando hacer la consulta. El valor por defecto son 5 segundos.



# Whois

## Whois

Es un protocolo TCP basado en petición/respuesta que se utiliza para efectuar consultas en una base de datos y permite determinar el propietario de un nombre de dominio o una dirección IP en Internet.

## Webs whois

<https://whois.icann.org> <https://who.is> <http://whois.domaintools.com>

# mtr

Realizar trazos de las comunicaciones de un equipo. Podemos apreciar el envío de paquetes a esta dirección de la LAN, además de:

- Paquetes perdidos. Loss%
- Paquetes enviados. Snt
- Velocidad del último paquete enviado. Last
- Velocidad media de los paquetes enviados. Avg
- El paquete que se ha enviado más rápido. Best
- El paquete más lento. Wrst.
- La calidad de los paquetes enviados, lentos o rápidos. StDev

## mtr --tcp

Obliga a mtr a enviar paquetes TCP en vez de UDP que sería lo normal.

## mtr -n

Vemos las direcciones ip de los nombres asignados en los DNS

## mtr -c

Indicamos el número de pings que queremos que haga.

## mtr -r

## webs mtr

<http://www.traceroute.org/#USA> <http://www.yougetsignal.com/tools/visual-tracert/> <http://www.ip-tracker.org/locator/ip-lookup.php>

- ip-tracker. Nos muestra información de la localización de la dirección IP, además nos aporta información sobre los servidores de nombres, su nombre inverso, su hostname en la red, información sobre el ISP, información atmosférica. También tiene otras

herramientas disponibles como, chequeo en listas negras o test de velocidad de la red.

<https://www.iplocation.net>

- iplocation. Nos muestra información sobre la dirección Ip o nombre de dominio que pongamos en el buscador de la web. Entre ellas esta la latitud y la longitud de la posición de esta. También nos ofrece otras herramientas como verificación y rastreo de direcciones de email, chequeos de proxy y calculadoras de subredes.

# ss

## ss -tun

Muestra las direcciones ip de las últimas conexiones realizadas con nuestro equipo a otras redes o direcciones ip. Con la opción -t mostramos los paquetes TCP, con la opción -u mostramos los paquetes UDP y con la opción -n muestra los puertos que se están utilizando.

## ss -l / ss -a

Si no añadimos ningún parámetro ss nos muestra todas las conexiones realizadas por el sistema, incluidas las conexiones locales de los sockets de UNIX.

- Opción -l. Nos muestra los sockets que están en escucha.
- Opción -a. Nos muestra todos los sockets disponibles.

## ss -p

Este parámetro únicamente funciona accediendo a ss como root, el parámetro -p nos muestra los procesos que están utilizando un socket.

## watch -n 2 ss-tun

Con el comando watch hacemos que la pantalla se refresque cada dos segundos.

# nc / netcat

- Opción -l. Indicamos a netcat que debe escuchar una conexión entrante por un puerto específico.
- Opción -p. Se utiliza para indicar el puerto que queremos que nc use.
- Opción -v. Modo verbal, que nos muestran las acciones o escuchas que ocurren en el momento.
- Opción -u. Para usar paquetes UDP en vez de TCP que son los que viene por defecto.
- Opción -w. Rechaza las conexiones que no se han realizado en el tiempo que determinemos, no afecta a la opción -l que siempre se mantendrá en escucha.

## SERVIDOR nc -l p 5588

Estamos a la escucha de la conexiones entrantes en el puerto 5588

## CLIENTE nc ipservidor 5588

Mandamos paquetes TCP a la ip del servidor al puerto 5588

## SERVIDOR nc -l -u -p 5555 < archivo

Estamos a la escucha de los paquetes UDP que entran en el puerto 5555, estos datos viene del archivo.

## CLIENTE nv -u -w 10 ipservidor 5555 > archivo

Enviamos paquetes UDP y siempre que la conexión sea inferior a 10 seg. guardamos la información en el fichero.

## SERVIDOR nc -l -p 6789

Estamos a la escucha en el puerto 6789

CLIENTE echo "QUIT" | nc -v ipserver 6789

SERVIDOR nc -v -l -p 8888 | dd of=archivo.iso

Estamos en la escucha en modo verbose del puerto 8888 y lo guardamos todo en una archivo.iso mediante el comando dd y su segundo paso el of.

CLIENTE dd if=/dev/sda | nc ipserver 8888

Mandamos por el puerto 8888 el flujo de datos para crear una copia de sda.

## Cabeceras HTTP con nc

printf "GET / HTTP/1.0/r/n/r/n" | nc www.redhat.com 80 > pagina.html cat pagina.html

## Resumen

nc -l v -p 50499. Ponemos el servidor en modo escucha ss -tunl | grep 50499. Comprobamos en el servidor que el servicio esta funcionando. nv -C localhost 50499. Desde el cliente nos conectamos.

En la terminal del primer comando podemos apreciar la conexión.

# curl

## Opciones curl

La opción -o nos guarda el resultado de la consulta con curl en un archivo.

La opción -O nos guarda el resultado de la consulta en un archivo con el nombre de la url.

La opción -C nos permite continuar una descarga que había sido cortada previamente.

La opción -I Captura únicamente las cabeceras HTTP, cuando la consulta es a un FTP o un archivo, curl solo nos muestra la cabecera.

La opción -s pone a curl en modo silencioso y no muestra las métricas o errores.

La opción -v modo verbose, se usa para debug y ver que está haciendo.

La opción -H nos muestra una cabecera extra cuando se envía una petición a un servicio HTTP.

La opción -L Si el servidor reporta que algunas de sus páginas han sido movidas de sitio, esta opción hace que curl siga la redirección.

## Cabeceras Principales

Con esta comando estamos pidiendo únicamente las cabeceras principales, como mínimo tenemos que tener la 200 OK.

```
curl -ILs https://www.twitter.com | grep "HTTP/"
```

## Enlaces principales

Nos muestra el enlace a la página [www.makeuseof.com](http://www.makeuseof.com) que viene desde twitter.

```
curl -sIL http://www.makeuseof.com | grep "^Location"
```

## Solo lo del grep

Al utilizar la opción -s únicamente lo que ha capturado el grep, en este caso la cantidad de seguidores.

```
Curl -s https://twitter.com/encampanya | grep -o "[0-9,]* Seguidores"
```

## Barra de progreso

Nos muestra una barra de progreso, en este caso una copa llenándose.

```
curl -s http://artscene.textfiles.com/vt100/wineglasses.vt | pv -L9600 -q
```

## Descarga de fichero

Nos guarda el archivo Linux-Voice-Issue-016.pdf

```
curl -O -C - https://www.linuxvoice.com/issues/016/Linux-Voice-Issue-016.pdf
```

## Bandeja entrada Motor RSS

Este comando nos muestra los mensajes que tenemos en nuestra bandeja de entrada a través del motor RSS, que

```
curl -u raulolmedom@gmail.com:password -s "https://mail.google.com/mail/feed/atom" | grep "Raúl" | wc -w
```

```
curl -u usuariGmail:contrasenyaGmail -s "https://mail.google.com/mail/feed/atom"
```

## Cabeceras de todas las imágenes de la web

Obtenemos la cabeceras con curl de todas las imágenes que tenga la web

Filtrando el resultado con grep y las expresiones regulares para que únicamente con los archivos que tengan el fo

Con cut cortamos y mostramos únicamente lo que hay después de las primeras comillas, que en este caso son los

```
curl http://concept-art.tumblr.com/ | grep -o 'src="[^\"]*.[png-jpg]"' | cut -d\" -f2
```

## Crear enlaces aleatorios

```
cat archivo.txt | curl -F 'clbin=<-' https://clbin.com
```

```
cat scripcurl.sh || curl -F 'clbin=<-' https://clbin.com
```

Hemos creado un enlace con el contenido del archivo, este se puede visitar desde cualquier navegador

## Consulta dominio especificando Ip

```
curl https://DOMAIN.EXAMPLE --resolve 'DOMAIN.EXAMPLE:443:192.0.2.17'
```



# vnstat y vnstati

vnstat -i enp0s3. Podemos ver los rangos de conexión mensuales y diarios, así como una estimación. vnstat -i -d enp0s3. Salen las conexiones que se han realizado por días. vnstat -i -m enp0s3. Salen las conexiones que se han realizado por meses. vnstat -i -w enp0s3. Salen las conexiones que se han realizado por semanas. vnstat -i -h enp0s3. Salen las conexiones que se han realizado por horas. vnstati -h -c 15 -ne -i enp0s3 -o hola.png. Nos ha creado un archivo png con la gráfica de la interfaz enp0s3 y sus conexión por horas, establecido con un rango de 15 Mbits por segundo.

# iperf

## SERVIDOR iperf -s p 5555

Ponemos en el servidor el servicio en escucha

## CLIENTE iperf -c ip\_server -f M

Ahora desde la máquina cliente estamos realizando la conexión, y el servidor nos guarda el ancho de banda y la tasa de transferencia de esta.

## iperf -i nº segundos

Nos ayuda a establecer el período de tiempo que pasará entre consulta y consulta de las tasas de comunicación

## iperf -t nº segundos

Nos ayuda a establecer el tiempo que durará cada conexión, por defecto son 10 segundos.

# nload / tcptrack / nethogs / tmux

## nload

Para activar. `nload enp0s3`. El paquete `nload` nos muestra estadísticas sobre el tráfico de entrada y salida en la tarjeta especificada. Podemos ver:

- Tráfico actual.Curr.
- La media del tráfico. AVG.
- El mínimo.
- El máximo.
- Y su máximo ancho de banda.

## tcptrack

Para activar. `tcptrack -i enp0s3`

- Opción p. Pausa la escucha o estadística.
- Opción s. Te permite elegir como ordenar las conexiones(rango de bytes desordenado).
- Opción q. Sale del modo interactivo.

## tcptrack -d

Nos muestra únicamente la conexiones que se hayan iniciado después de que hayamos ejecutado `tcptrack`.

## nethogs

Para activar `nethogs enp0s3`.

Ordenar por paquetes recibidos con la tecla `r`, ordenar por paquetes enviados la tecla `s`, y la tecla `q` para salir.

# tmux

Software que nos permite abrir terminales dentro de un mismo terminal.

- CTRL + B -> %. Crea una ventana paralela.
- CTRL + B -> ". Crea una ventana inferior.
- CTRL + B -> cursores. Nos desplazamos por las ventanas.

Enlace para obtener más información acerca de tmux <https://tmuxcheatsheet.com>

# nmap

Software para escanear sistemas o equipos.

Para seleccionar los equipos a escanear a diferentes posibilidades:

- \*Un ip de host individual o diferentes IP separadas por espacios
- \*Un rango de direcciones IP indicado por guiones. Ex: 192.168.10.20 - 192.168.10.50
- \*Un conjunto de ip de diferentes redes, indicados por comas. Ex: 192.168.10.1,10.0.0.4,11.25.10.30
- \*Una ip de red con su máscara correspondiente. Ex: 192.168.10.0/24
- \*Un hostname

## Opciones más importantes

- Opción -p. Indicamos los puertos que queremos escanear. -p 5555. -p 5555,3853,4543, -p 5555-5559
- Opción -sn. Para ver si los equipos están presentes en la red y sin escanear ningún puerto.
- Opción -PO. Si ya sabemos que la víctima esta en la red y solo queremos escanear unos puertos concretos.
- Opción -A. Para averiguar el S.O., los programas que utiliza la víctima y sus versiones. Se debe usar como root
- Opción -T{0-5}. Para establecer el retardo entre paquete y el siguiente, hace el escaneo más o menos silencioso.
- Opción -iL fichero.txt. Desde donde cogeremos las IP y nombres de las víctimas.
- Opción -oN fichero.txt. El resultado del escaneo se guarda en un fichero. -oX salida XML o -oG salida para grep.
- Opción --packet-trace: Muestra la trama de todos los paquetes enviados por nmap cuando realiza su trabajo.
- Opción -n. No hace resolución de nombres.
- Opción -v. Modo verbal.

## Averiguar las máquinas que hay en una red

```
nmap -sn 192.168.10.0/24
```

## Averiguar puertos abiertos

```
nmap -p 1-10000 -Pn --reason ip_victima
```

# Averiguar programas y versiones

```
nmap -p 1-10000 -Pn -A ip_victima
```

Si añadimos el parámetro -T5. Aumenta la velocidad del escaneo ya que estamos usando su versión más rápida y

## Escaneo Puertos

- -sS. Analiza los paquetes SYN/Connect()
- -sU. Analiza los paquetes UDP

## Escaneo Puertos -> Salida XML

```
nmap -T50 -PO -p 0-10000 -oX - 192.168.1.1 -> fichero.xml
```

Para UDP

```
nmap -T5 -PO -sU -p 0-10000 -oX - 192.168.1.1 -> fichero.xml
```

# nmcli

Network Manager para terminal.

## Activar/Desactivar Wifi

```
nmcli radio wifi [on/off]
```

## Estado interfaces de red

```
nmcli device status
```

## Estado de una interfaz específica

```
nmcli device show enp0s3
```

## Conexión/Desconexión de la tarjeta de red

```
nmcli device {disconnect | connect} enp0s3
```

## Puntos de acceso WIFI

```
nmcli device wifi list
```

## Escaneo en busca de puntos de acceso wifi

```
nmcli device wifi rescan
```

## Conexión a un punto de acceso

```
nmcli device wifi connect xeill
```

# Opciones nmcli

- Opción -t. Nos quita la tabulación cuando nmcli muestra información.
- Opción -p. Nos da una vista mejora de la información en caso de tenerla.
- Opción -m. También tiene que ver con la vista que queremos que saque nmcli. Mediante tabulaciones o multilínea. Multilínea es la que viene predefinada.
- Opción -f. Nos permite elegir los campos que vamos a mostrar. EX. nmcli -t GENERAL,DHCP4 device show enp0s3. Si no introducimos bien los campos nmcli nos muestra una lista de los campos que podemos utilizar.

# Cambios con nmcli

```
nmcli connection add con-name pepito type ethernet ifname "*"
(per defecte es crea fent servir DHCP)
nmcli connection modify pepito ipv4.addresses 192.168.100.100/24
nmcli connection modify pepito ipv4.gateway 192.168.100.10
nmcli connection modify pepito ipv4.dns "8.8.8.8 8.8.4.4"
nmcli connection modify pepito +ipv4.dns "192.168.15.10"
nmcli connection up pepito
nmcli connection show
nmcli connection show pepito | grep "ipv4"
```

# nmtui

Network Manager Text User Interface.El programa nmtui nos facilita una interfaz gráfica para la configuración de las interfaces de red:

- pudiendo asignar IP estática o dejar que la interfaz la obtenga por DHCP.
- desactivar IPv6.
- asignar DNS.
- modificar la MTU.
- activar y desactivar interfaces.
- asignar hostname a la máquina anfitriona.



# ssh

Existe una forma de loguearnos en servidores SSH que implica no tener que escribir usuario/contraseña cada vez y que lo hace más segura. Se trata de utilizar una pareja de claves pública/privada de usuario. La idea es que cada usuario genere su propia pareja de claves y coloque su clave pública en el servidor.

## Crear una pareja de claves

```
ssh-keygen -b 1024 -t dsa
```

Al finalizar este comando nos crea dentro de la carpeta de usuario, una carpeta llamada `.ssh/`, que en su interior alberga la clave pública y privada que acabamos de crear, además de los hosts que ya se conectaron con esta equipo.

Si quisiéramos cambiar el passphrase que nos pide al crear la clave tenemos las siguientes opciones:

- Para DSA. `ssh-keygen -f id_dsa -p.`
- para RSA. `ssh-keygen -f id_rsa -p.`

## Pasar clave pública a el servidor

Permisos de la carpeta `.ssh` 700 Permisos del archivo `authorized` 600

1. `cat ~/.ssh/id_dsa.pub | ssh pepito@servidor "cat - >> ~/.ssh/authorized_keys"`
2. `scp ~/.ssh/id_dsa.pub pepito@servidor:~/.ssh/authorized_keys`
3. `ssh-copy-id -i ~/.ssh/id_dsa.pub pepito@servidor.`

## Configuración servidor

- `PermitRootLogin without-password` (o “no”, directament)
- `PubkeyAuthentication yes` (activem el sistema d'autenticació per claus)
- `PasswordAuthentication no` (si no es vol usar més el sistema d'autenticació per contrasenyes)
- `RSAAuthentication yes` (si a més de poder utilitzar l'algoritme DSA volem permetre claus RSA)

- AuthorizedKeysFile %h/.ssh/authorized\_keys (indica l'ubicació i nom de l'arxiu authorized\_keys)
- Port. Indicamos el número del puerto de escucha del sshd.
- PermitEmptyPasswords. Permite establece conexión ssh sin ninguna contraseña. No recomendado por el creador del programa.
- PrintMotd i Banner.

El primero especifica a sshd si debe imprimir el contenido del archivo /etc/motd Mensaje que se mostrará cuando el usuario se conecta al servidor.  
El segundo nos da la posibilidad de enviar el contenido de un archivo especificado antes de permitir la conexión.

- AllowUsers o DenyUsers. Ssh puede acceder a una lista para permitir o no el acceso de usuarios. Estos usuarios pueden ser aceptados o denegados por defecto.
- ForceCommand
- LoginGraceTime. Es el tiempo que le damos al usuario para que complete el proceso de logueo. Por defecto son 120 segundos, y si ponemos el valor a 0 es tiempo infinito.
- MaxAuthTries. Especifica el número máximo de conexiones permitidas por usuario. Por defecto es 6.
- Match User i Match Host i Match Address
- StrictHostKeyChecking. Esta opción nos permite establecer en el servidor si preguntaremos o no justo la acción anterior de anotar en el archivo del cliente known\_hosts la clave pública para futuras conexiones. Si la establecemos como no, directamente el servidor incluirá dicha clave sin preguntar al cliente.

# rsync

## VM con NAT en Virtualbox

El modo RED NAT en VirtualBox es muy interesante porque nos permite acceder a Internet desde cada máquina virtual y además nos permite tener una red interna entre ellas. Con direcciones IP por defecto de la red 10.0.2.0/24 y dadas automáticamente por el propio VirtualBox.

### Redireccionamiento VirtualBox.png

Hemos añadido dos máquinas con NAT, para poder iniciar una comunicación con el equipo anfitrión por SSH debemos especificar un puerto: Vamos a la configuración de nuestra tarjeta de red -> opciones avanzadas -> y pulsar el botón de reenvío de puertos.

Una vez en la ventana correspondiente creamos los redireccionamientos:

Nombre VM	TCP / UDP	Ip anfitrión	Puerto	Ip invitado	puerto
HostB	TCP	127.0.0.1	2222	10.0.2.15	22
HostA	TCP	127.0.0.1	2223	10.0.2.15	22

Una vez creados comprobamos que los puertos están a la escucha y conectamos con las dos máquinas por ssh:

```
nc -nap | grep 2222
ssh -p 2222 usuario@127.0.0.1
ssh -p 2223 usuario@127.0.0.1
```

## Servidor Rsync

En la máquina A vamos a instalar el servicio Rsync.

```
*apt update && apt install rsync
*Crear fichero de configuración en: /etc/rsyncd.conf
```

Contenido del fichero rsyncd.conf

```
port = 1200 #No es necesario si no queremos cambiar el puerto por defecto 873 TCP
max_connections = 2 #Nº máximo de conexiones
timeout = 300 #Todas las operaciones en las que no haya ningún tipo de operación de I/O durante 300 segundos
[casa]
```

```
path = /home/usuario #Especificamos el directorio donde permitos las conexiones. Interesante opción ya que pod  
read_only = false  
list = true
```

Reiniciamos el servicio

```
systemctl restart rsync
```

## Opción host\_allow

Nos permite establecer una patrones para las conexiones, ip, mac, nombre host, etc. Si la conexión no cumple algunos de las marcas establecidas esta es rechazada.

```
hosts_allow = 192.168.0.0/24
```

## Formas de ejecución rsync

Para saber que módulos comparte el servidor, siempre que tenga la opción list=true.

```
rsync rsync://ip_server:port
```

Para ver el contenido de un módulo compartido llamado "casa".

```
rsync rsync://ip_server:port/casa
```

Para descargar todo el contenido de casa en la carpeta /web/prueba

```
rsync -a rsync://ip_server:port/casa /web/prueba
```

Para descargar únicamente un fichero

```
rsync -vP rsync://ip_server:port/casa/fichero /web/prueba
```

Para subir la carpeta web ella includa

```
rsync -a /web rsync://ipserver:port/casa
```

Para subir solo el contenido de la carpeta web

```
rsync -a /web/ rsync://ipserver:port/casa
```

El parámetro -a equivale a un conjunto de otros parámetros más específicos:

- -r Copia recursiva
- -l Copia los enlaces como enlaces, no los ficheros reales a los que apuntan
- -p Mantiene los permisos de los ficheros originales
- -t Mantiene la fecha de la última modificación de los ficheros originales en los ficheros copia, si no se especifica esta fecha será la fecha de copia.
- -o i -g. Mantiene el propietario y el grupo del propietario, si no se especifica el propietario y el grupo serán los usuarios que crean la copia.

## Otros parámetros

- -v Verbose mode.
- -n Modo simulación.
- -z Comprimos los ficheros durante la transferencia. Ahorra ancho de banda pero fuerza la CPU.
- -P Combinación de dos parámetros.
  - --progress muestra una barra de progreso.
  - --partial Para poder continuar una transferencia interrumpida.
- --bwlimit=no Establecer un ancho de banda.
- --delete Si al copiar el contenido de origen se detecta que el destino hay algún elemento(fichero o directorio) que no esta presente en el origen, LO BORRA DE DESTINO.
  - -b Si no se quiere ser tan radical con --delete y borrar los ficheros que sobran en el destino, con la opción -b le decimos que estos ficheros que nos sobran no los borre, sino que los guarde en la carpeta que le indicamos.
- --exclude="patron" No realiza la transferencia de los ficheros que coincidan con el patrón.

```
rsync --exclude="*.bak" o rsync --exclude="fichero_exclusiones.txt"
```

## Seguridad

Podemos obligar a los usuarios a conectarse con usuario y contraseña, esta le añade un nivel extra de seguridad al servicio Rsync. Estos usuarios y contraseñas son propios de Rsync, para crearlos los hemos de añadir al fichero /etc/rsync.secrets, el nombre del fichero no es obligatorio pero su contenido tiene que ser como este:

```
usuario1:pass1
usuario2:pass2
usuario3:pass3
```

El propietario de este fichero debe ser root y tener permisos 600 para que nadie pueda ver lo que contiene. Una vez configurado el servidor, se debe crear un archivo de secretos con el siguiente contenido:

```
secrets file = /etc/rsyncd.secrets  
auth users=usuario1,usuario2
```

Para conectarnos desde el cliente con un usuario específico:

```
rsync rsync://usuario1@ipserver:port  
rsync rsync://usuario2@ipserver:port/modulo
```

# Iptables

## OPCIONES IPTABLES

Para limpiar la reglas y tablas anteriores

```
iptables -F iptables -Z
```

Inicio de política restringida

```
iptables -P INPUT DROP iptables -P OUTPUT DROP iptables -P FORWARD DROP
```

Bloquear todo el tráfico entrante

```
iptables -A INPUT -i eth0 -j DROP
```

Aceptamos los paquetes que usa ping (ICMP)

```
iptables -A INPUT -p icmp -j ACCEPT
```

Aceptamos ssh desde un host y rsync desde cualquier red

```
iptables -A INPUT -p tcp -s IP_HOST --dport 22 -j ACCEPT iptables -A INPUT -p tcp --dport 12000 -j ACCEPT
```

Aceptamos contestación desde el servidor

```
iptables -A OUTPUT -m conntrack --cstate RELATED,ESTABLISHED -j ACCEPT
```

# Mostrar todas las reglas

```
iptables -L
```

# Registro y expulsión de todas las conexiones ICMP

```
iptables -A INPUT -p icmp --icmp-type echo-request -s 0.0.0.0 -j LOG --log-prefix "envio de paquete ICMP"
```

```
iptables -A INPUT -P icmp --icmp-type echo-request -s 0.0.0.0 -j REJECT
```

# SSH de Lunes a Viernes de 08:00 a 20:00

```
iptables -p tcp --dport 22 --days Fri,Sat,Sun --timestart 0800:00 --timestop 20:00:00 -j ACCEPT
```

# Descartamos más de 3 intentos de conexión

```
iptables -A INPUT -p tcp --dport 22 -m connlimit --connlimit-above 3 -j DROP
```

# Descartamos todas las conexiones con las máquinas que realicen más de 5 peticiones por minuto

```
iptables -A INPUT -p tcp -m limit --limit 5/m --limit-burst 5 -j DROP
```

# Regla PAT el puerto 22

```
iptables -A PREROUTING -p tcp --dport 22 -j REDIRECT --to-port 15000
```

# Activar ip-forward



Utilizamos el siguiente comando para establecer el `ip_forward=1` y de esta manera dejarlo activado: `sysctl -w net.ipv4.ip_forward=1`

## Masquerade

```
iptables -t nat -A POSTROUTING -o enp0s3 -j MASQUERADE
```

## #Reglas para escuchar mostrar las conexiones del hostB a la red del anfitrión de las #máquinas virtuales

```
iptables -t nat -A PREROUTING -i enp0s8 -p tcp --dport 678 -j DNAT --to-destination 192.168.10.6:5678
```

```
iptables -t nat -A POSTROUTING -o enp0s8 -p tcp --dport 5678 -d ip_hostB -j SNAT --to-source 192.168.10.6
```

## Por defecto DROP, permitir Servicio SSH

```
iptables -A INPUT -i eth0 -p tcp --dport 22 -m conntrack --ctstate NEW,ESTABLISHED -j ACCEPT  
iptables -A OUTPUT -i eth0 -p tcp --sport 22 -m conntrack --cstate ESTABLISHED -j ACCEPT
```

## Bloquear por intentos de conexión

```
iptables -A INPUT -i eth0 -s 10.0.1.0/8 -m limit --limit 5/m --limit-burst 7 -j LOG --log-prefix "Ojo peligro"
```

```
iptables -A INPUT -i eth0 -s 10.0.1.0/8 -j DROP
```

## Descartar paquetes TCP null

```
iptables -A INPUT -p tcp --tcp-flags ALL NONE -j DROP
```

## Descartar paquetes "Xmas"

```
iptables -A INPUT -p tcp --tcp-flags Xmas -j DROP
```

## Descarte paquete peligrosos de cierre de conexiones

```
iptables -A INPUT -p tcp --tcp-flags SYN,ACK,FIN,RST RST -m limit --limit 1/s -j DROP
```

## Evitar más de un respuesta en el mismo segundo

```
iptables -A INPUT -p icmp --icmp-type echo-request -m limit -limit 1/s -j DROP
```

## Algunos ejemplos

### Bloquear tráfico de una tarjeta y una dirección concreta

```
iptables -A INPUT -i eth0 -s 192.68.10.11 -j DROP
```

### Bloquear tráfico entrante de una tarjeta y puerto concreto

```
iptables -A INPUT -i eth0 -p tcp --dport 22 -j DROP
```

### Bloquear todo el tráfico de una tarjeta y que vaya destinado a puerto concreto

```
iptables -A INPUT -i eth0 -p tcp --dport 22 -j LOG --log-prefix "Acceso al puerto 22"
```

# conntrack

## conntrack -L

Hace un seguimiento permanente de todas las conexiones

## Seguimiento conexiones al puerto 22

```
conntrack -L -p tcp --dport 22 --state ESTABLISHED
```

## Total de conexiones

```
conntrack -C
```

## Mostrar log

```
conntrack -L
```

## Solo nuevas líneas del log

```
conntrack -E -e NEW
```

## Estadísticas

```
conntrack -S
```

## Configuración

Si queremos saber el máximo de conexiones que permitimos para las líneas conntrack debemos acudir al archivo: **/proc/sys/net/netfilter** o en versiones más antiguas **/proc/sys/net/ipv4/netfilter**