

CAPÍTULO 7

Introducción a las redes TCP/IP

TCP/IP

Una de las características más importantes de Linux es su cualidad como servidor de redes. La red en Linux (y el propio Linux de forma interna) habla el mismo idioma que Internet, lo que hace que el hecho de dar servicios de Internet y acoplar una red de área local no suponga prácticamente ningún esfuerzo, ya que en el proceso no se necesitan máquinas traductoras. Este lenguaje se conoce de forma genérica como TCP/IP (de las siglas en inglés de Protocolo de Control de Transmisión/Protocolo de Internet); en realidad son un conjunto de protocolos agrupados para crear una comunicación más fluida por la red, cada uno de ellos realizando un trabajo muy específico.

El funcionamiento de TCP/IP se basa en el uso de unas direcciones, denominadas direcciones IP, para identificar de qué máquina procede cada mensaje y a qué máquina se dirigen. Estas direcciones tienen el formato: ###.###.###.###, donde cada grupo de tres dígitos (denominado octeto ya que se puede representar por un Byte) puede estar comprendido entre 0 y 255; de ella una parte se dedica a la dirección de red y el resto identifica a la máquina dentro de esa red. El número de cifras de la dirección dedicada a cada uno de estas componentes (red y máquina) varía de un tipo de red a otra. En cierto modo se asemeja a un número de teléfono, donde las primeras cifras forman el prefijo de zona y el resto el número de abonado dentro de la zona.

Para separar las dos partes se utiliza la máscara de red, que no es más que un número con la misma estructura de dirección IP pero con un significado distinto. Así, se distinguen varios tipos de direcciones que definen otras tantas clases de redes, cada una con una máscara de red distinta:

A:	0.0.0.0 a 127.255.255.255	255.0.0.0
B:	128.0.0.0 a 191.255.255.255	255.255.0.0
C:	192.0.0.0 a 223.255.255.255	255.255.255.0

Existen dos tipos de redes más (tipo D, de 224.0.0.0 a 239.255.255.255, y

tipo E, de 240.0.0.0 a 247.255.255.255, pero no se utilizan)

En una red de tipo A el primer octeto numera a la red, mientras que los otros tres numeran las máquinas. En redes de tipo B sin embargo se reservan dos octetos para la red y dos para las máquinas. En las de tipo C son tres los octetos para numerar la red y uno para la máquina. Esto puede observarse por la máscara propia de cada tipo de red.

Hay una dirección especial compartida por todas las máquinas que utilizan TCP/IP; es la dirección 127.0.0.1 y se denomina dirección de circuito cerrado o dirección de lazo local; siempre es una referencia a la propia máquina en la que se trabaja y es comúnmente conocida como "localhost".

En toda red hay un par de direcciones "reservadas", que no pueden utilizarse para numerar a ninguna máquina. Estas dos direcciones se conocen con el nombre de "dirección de red" (que siempre es la menor de toda la red) y "dirección de difusión" o "dirección de broadcast" (que siempre es la última). La primera sirve para nombrar e identificar a la red como un bloque, mientras que la dirección de difusión se utiliza para enviar datos de manera simultánea a todas las máquinas de la red.

De esta forma, podemos ver que existen 126 redes distintas de tipo A cada una con algo más de un millón y medio de máquinas posibles. Existirán del mismo modo unas 16000 redes de tipo B, cada una con 65000 máquinas. Por último, hay algo más de dos millones de redes de tipo C en las que caben 254 máquinas en cada una.

Para poder utilizar una IP es necesario solicitarla, y sólo se conceden a organismos o empresas capacitadas para ello (por ejemplo los proveedores de internet); cada vez es más difícil disponer de una ya que se está llegando a una situación en la que casi no quedan IP disponibles, debido a que en el actual sistema de numeración se desaprovechan muchas direcciones (si tenemos una red de tipo A y poseemos 700000 máquinas estamos desaprovechando todavía 16 millones de direcciones que nadie puede utilizar. Esto se ha solucionado implantando un sistema de numeración de 64 bits (en lugar de los 32 actuales); este sistema, a parte de ser virtualmente inagotable (equivaldría a tener unos 200 aparatos por metro cuadrado cubriendo toda la superficie de la tierra) es compatible con el actual. Ésta y otras mejoras se introducen en la versión 6 del protocolo IP (conocida como IPv6, la actual es IPv4), que tímidamente se está empezando a implantar en algunos sistemas (Linux es totalmente compatible con él desde hace tiempo).

Por otro lado, la estructura de la comunicación entre dos máquinas responde a un modelo de capas, una sobre otra, de manera que el conjunto de tareas se divide en partes que son ejecutadas en una determinada capa. Cada capa se comunica con una similar en el otro extremo de la comunicación

intercambiando una serie de datos, que son utilizados por la capa superior; cada capa además posee una información de control que es entendida únicamente por su capa pareja y que es la que permite que se produzca el intercambio de datos.

En TCP/IP existen cuatro capas diferenciadas; la primera es la capa de red local, que es la encargada de llevar la información de una máquina a otra, resolviendo los problemas que puedan ocurrir en la transmisión (ya sea por cable, inalámbrica, óptica, ...). Los elementos que residen en esta capa son los controladores de dispositivos de red, como la tarjeta Ethernet o el módem.

Sobre ella se encuentra la capa internet o “entre redes”, que es dónde reside IP. Es la responsable de decirle a los datos dónde tienen que ir para llegar a su destino, para lo que utiliza las direcciones IP (lo veremos un poco más adelante); el envío de datos se realiza a una máquina accesible directamente de la red.

La capa de transporte es la primera que conoce la transmisión extremo a extremo, no a máquinas próximas como hace la capa de red. Aquí es donde trabaja TCP, que se encarga de fragmentar los paquetes en trozos más pequeños que son enviados junto con información para su ensambladores. TCP se ocupa también de servicios que requieren una conexión ininterrumpida con comprobaciones de los errores y recuperación ante ellos. Para servicios que no necesitan controlar si hubo o no un error o que no necesitan de una conexión establecida permanentemente con el receptor se utiliza el protocolo UDP (Protocolo de Datagrama de Usuario), más rápido que TCP en ambientes fiables (como una red de área local, por ejemplo).

Por último tenemos la capa de Aplicación, que es donde residen los diferentes programas que utilizan Internet; cada una de estas aplicaciones ofrece unos servicios al usuario (o a otros programas), para lo que cual necesitan utilizar uno de los protocolos de la capa de transporte, ya sea TCP o UDP. Por ejemplo los servicios FTP, POP3 y SMTP utilizan TCP, DNS y NFS usan UDP y HTTP utiliza ambos. Para poder conocer a qué aplicación se destinan los datos se utiliza un número de puerto, de manera que la pareja dirección IP/puerto determina de forma única la aplicación a la que se destinarán los datos.

No te preocupes si no conoces el significado de todas estas siglas, las veremos en detalle más adelante; de todas formas existe una lista exhaustiva de los servicios existentes y los puertos y protocolos que utilizan en el archivo /etc/services. Lo que debes tener muy claro es que TCP y UDP poseen puertos independientes, es decir, cada uno tiene su propia numeración.

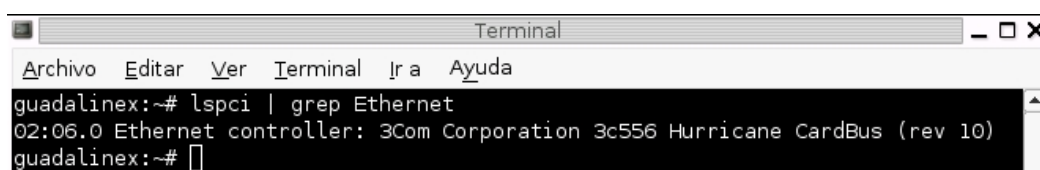
Configuración del Hardware

Para la interconexión física de ordenadores se emplea algún tipo de tecnología de red de área local; la más común con diferencia son las redes Ethernet. Para construir una red de este tipo cada ordenador debe poseer una tarjeta de red o interfaz de red; esta interfaz posee una dirección única que viene grabada de fábrica en cada tarjeta, por lo que se suele llamar dirección física; en redes Ethernet se compone de 6 bytes, que se suelen representar en hexadecimal (por ejemplo 06:00:1b:4c:48:33).

Otro mecanismo muy común para conectarse a una red es mediante una conexión punto a punto a otro ordenador (típicamente con un módem); en este caso la red local está formada únicamente por dos ordenadores. La tecnología equivalente para este tipo de conexiones se denomina PPP (Protocolo Punto a Punto). Para este tipo de conexión existe un icono en el escritorio llamado "Conexión a Internet", el cual posee instrucciones precisas para el caso de módem analógico, módem USB o router USB, por lo que el resto del capítulo nos centraremos en el caso de redes locales.

Para configurar una conexión de red es necesario ejecutar dos tareas básicas: configurar la interfaz de red y configurar el TCP/IP.

La configuración de la interfaz consiste básicamente en cargar el driver apropiado para la tarjeta o tarjetas que tengamos. Estos módulos se cargan como cualquier otro módulo del núcleo (generalmente se hace automáticamente en el arranque); para saber qué tipo de tarjeta de red tenemos podemos recurrir a la salida del comando "lspci".



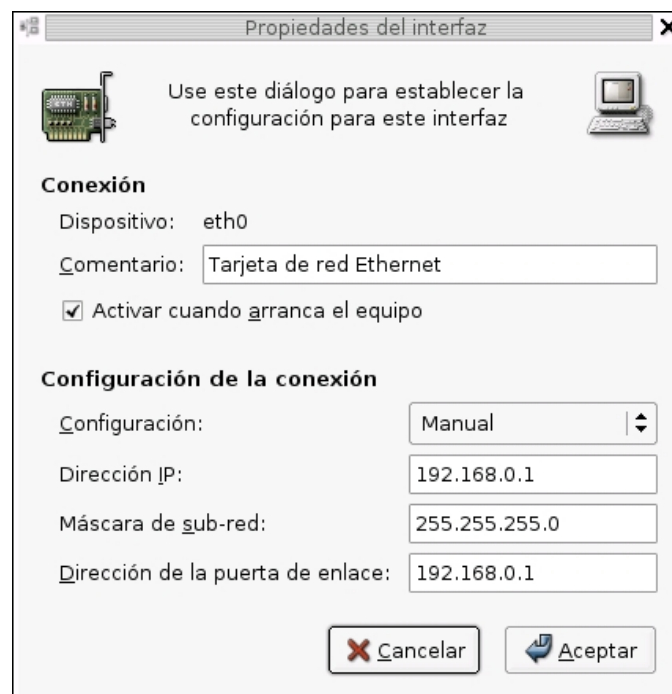
```
Terminal
Archivo  Editar  Ver    Terminal  Ir a  Ayuda
guadalinux:~# lspci | grep Ethernet
02:06.0 Ethernet controller: 3Com Corporation 3c556 Hurricane CardBus (rev 10)
guadalinux:~#
```

A continuación debemos conocer el módulo necesario para esta tarjeta; para este caso resulta ser el 3c59x (se suele encontrar esta información con ayuda de cualquier buscador de Internet), de manera que para cargar el módulo tan sólo hay que poner "modprobe 3c59x" y el módulo será cargado en memoria. Cada módulo cargado asignará una vez configurado un nombre de dispositivo a la tarjeta, empezando por eth0 y siguiendo por eth1, etc (para equipos con varias tarjetas, si no estará sólo disponible eth0).

Respecto a la configuración de TCP/IP, disponemos de una herramienta gráfica en Aplicaciones -> Herramientas del Sistema -> Panel de Control -> Red.

En primer lugar ponemos el nombre que la máquina tendrá en la red; este valor se guarda en el archivo `/etc/hostname`. Los siguientes apartados se refieren a la inclusión de nuestro Linux en una red con máquinas Windows y lo trataremos en un capítulo posterior.

La siguiente pestaña es quizá la más importante, ya que desde aquí se controlan las distintas interfaces de red de que disponemos; podemos añadir, borrar, activarla o desactivarla, y la principal, modificar sus propiedades, como su dirección IP, máscara de red y la puerta de enlace. Esta última es la dirección IP de una máquina que hace de puente entre nuestra red y otra, de manera que cuando intentemos comunicar con una máquina que no esté en nuestra red los paquetes se encaminen a través de esta máquina; el caso típico es el de una máquina o router que se utiliza para dar conexión a Internet a toda una red local (Internet no es más que una conexión entre muchas redes).



Los valores que se muestran no están elegidos al azar. En principio una red local aislada del resto de redes de Internet podría tomar cualquier valor para sus direcciones IP (aunque todas dentro de la misma red), ya que esta red no va a interferir con el resto de redes; sin embargo es costumbre utilizar una serie de rangos IP para configurar redes locales (que no se utilizan para máquinas conectadas directamente a Internet), rangos que se reservan precisamente para este uso. Existen 3 rangos disponibles: 10.0.0.0 a 10.255.255.255, 172.16.0.0 a 172.31.255.255, y 192.168.0.0 a 192.168.255.255 (como puedes observar hay un rango de tipo A, otro de tipo B y otro de tipo C, para que escojamos según nuestras necesidades).

También se puede configurar si la interfaz se activará al iniciar la máquina o si se realiza una selección automática de los valores en el menú

despegable “Configuración”; aquí las opciones son manual, DHCP o BOOTP (para utilizar las dos últimas es necesario que en la red existan sendos servidores dedicados a estos servicios).

Aunque ya podemos comunicarnos con el resto de máquinas, nos falta todavía activar una característica para hacer nuestros movimientos por la red más cómodos. Estarás de acuerdo conmigo que recordar las IP de las máquinas a las que queremos conectarnos es prácticamente imposible; por ello disponemos de dos características que nos resuelven esta tarea: los anfitriones y los servidores de nombres.

La lista de anfitriones es simplemente una lista que asocia a una IP un nombre, de manera que podemos introducir las IP de las máquinas de nuestra red con sus respectivas IP, y a partir de entonces nos podemos referir a esa máquina con ese nombre. Hay que tener en cuenta que esta lista debe introducirse en cada máquina de la red que se quiera que la utilice; en realidad sólo es necesario hacerlo una vez, ya que la lista se almacena en el archivo `/etc/hosts`, con lo que se puede copiar a todas las máquinas de nuestra red. Puedes ver que hay una entrada para “localhost” (el resto es necesario para poder usar IPv6).

Ahora bien, estos nombres sólo tienen ámbito local. Si queremos acceder a máquinas de fuera de nuestra red la lista debería ser bastante larga; en su lugar se utilizan unos servidores de Internet dedicados a almacenar esas extensas listas; son los denominados “Servidores de Nombres de Dominio” o DNS. Sus IP deben ser proporcionadas por tu proveedor de Internet (cada proveedor tiene las suyas, aunque en realidad puedes usar las de cualquiera). Simplemente puedes añadirlos desde la pestaña que reza “DNS”.

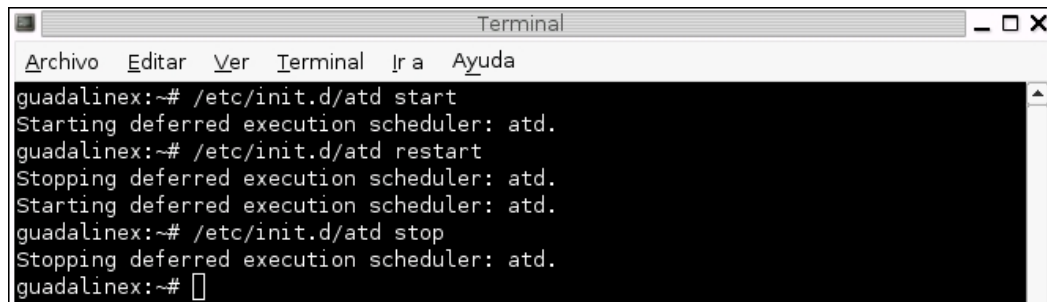
Servicios y control de acceso

Ya sabes que existen una serie de servicios que pueden estar ejecutándose en tu máquina y que ofrecen algún tipo de utilidad; algunos de ellos están pensados para dar servicio a otras máquinas de la red además de la propia. Ya sabes cómo activar y desactivar estos servicios desde el entorno gráfico, pero ahora lo veremos desde la consola.

Típicamente los servicios (tanto de red como los demás) se controlan mediante el uso de un script de texto localizado en `/etc/init.d`; ejecutando este script se puede iniciar el servicio, detenerlo o reiniciarlo (por ejemplo después de realizar alguna modificación en su configuración), tan sólo es necesario

conocer el nombre de ese script.

Ya conoces el demonio atd, encargado de la ejecución programada de comandos; para iniciarlo es necesario invocar desde la línea de comandos: `/etc/init.d/atd start`, para detenerlo: `/etc/init.d/atd stop`, y para reiniciarlo `/etc/init.d/restart`.

A screenshot of a terminal window titled "Terminal". The window has a menu bar with "Archivo", "Editar", "Ver", "Terminal", "Ir a", and "Ayuda". The terminal shows the following commands and output:

```
guadalinux:~# /etc/init.d/atd start
Starting deferred execution scheduler: atd.
guadalinux:~# /etc/init.d/atd restart
Stopping deferred execution scheduler: atd.
Starting deferred execution scheduler: atd.
guadalinux:~# /etc/init.d/atd stop
Stopping deferred execution scheduler: atd.
guadalinux:~#
```

Este funcionamiento de los servicios recibe el nombre de “standalone” ya que el demonio estará a la espera de que se necesiten sus servicios. Pero existe otra forma de tener estos servicios en funcionamiento conocida como activación “bajo demanda”; este modo de funcionamiento está reservado a los servicios de red y se basan a su vez en un servicio llamado “inetd”.

El programa inetd es un demonio (conocido a veces como “superdaemon” o superdemonio) que escucha peticiones en muchos puertos a la vez; cuando detecta actividad en alguno de los puertos arranca el servidor correspondiente al servicio solicitado. Este funcionamiento es conveniente por dos razones: evita tener en continuo funcionamiento un gran número de servidores que sólo se emplean ocasionalmente, y permite conceder o denegar el acceso en función del origen de la conexión antes de iniciar el servidor correspondiente.

Este servicio se configura en el archivo `/etc/inetd.conf`. Cada línea se utiliza para iniciar un servicio; si consultas este archivo encontrarás estas dos líneas:

```
smtp stream      tcp    nowait    mail  /usr/sbin/exim exim -bs
ftp  stream      tcp    nowait    root  /usr/sbin/tcpd
/usr/sbin/in.ftpd
```

La primera línea configura el servidor de correo electrónico de correo saliente o MTA (siglas en inglés de Agente de Transporte de Correo). El primer valor es el nombre del servicio tal y como aparece en `/etc/services`. Los dos siguientes indican el tipo de conexión y el protocolo a utilizar, y generalmente valen “stream tcp” o “dgram udp”, según que el servicio utilice un flujo (stream) TCP o datagramas UDP. El campo wait/nowait sólo tiene validez para datagramas e indica si hay que esperar a que se cierre la conexión abierta para atender otras peticiones.

A continuación se encuentra el usuario (y eventualmente el grupo) a través del cual se ejecutará el servidor; recuerda que vimos que existían usuarios “ficticios” que no se correspondían con una persona física sino que eran utilizados por programas y demonios para funcionar. En este caso el servicio smtp tomará la personalidad del usuario y grupo “mail”.

Para finalizar se encuentra la ruta completa del programa ejecutable para ese servicio seguido en su caso de los argumentos necesarios. En general no es necesario modificar este archivo a mano, puesto que al instalar el servidor correspondiente añadirá la línea adecuada en el archivo inetd.conf.

En ocasiones no aparece el programa ejecutable del servidor directamente sino que aparece “/usr/sbin/tcpd” antes de él, como ocurre en la segunda de las líneas; de esta forma se puede realizar el control de acceso que veremos a continuación. Dicho control de acceso permite o deniega el acceso a los servicios en función del origen de la conexión; este mecanismo se configura en los archivos /etc/hosts.allow y /etc/hosts.deny, donde se especifican unas sencillas reglas que permiten aceptar o rechazar conexiones entrantes. Además de los servicios activados mediante inetd hay algunos servidores que siempre utilizan este método de control de acceso (como sshd), conocido como TCP wrappers (cubiertas o envolturas para TCP).

Estos dos ficheros tienen una lista de la forma:

demonio : cliente

“Demonio” es el nombre del ejecutable, no el nombre del servicio (cuidado con esto, es fuente de numerosos quebraderos de cabeza) y el cliente es el origen al que se le permite o deniega el acceso a ese servicio.

Como demonio se pueden poner uno o más nombres separados por espacios o comas, y pueden usarse comodines como “?” (coincidencia con un carácter) o “*” (coincidencia con cualquier grupo de caracteres); también puede usarse la palabra “ALL” para indicar “cualquier servidor”. El cliente por su parte puede ser una o más direcciones IP, redes o nombres de máquinas separadas por espacios o comas; se pueden usar las palabras LOCAL (para especificar conexiones desde la misma máquina) o EXCEPT, para escribir excepciones.

El funcionamiento de ambos ficheros es muy sencillo; cuando el servicio arranca, se comprueba el contenido de /hosts.allow en busca de alguna regla que coincida; si se encuentra se permite el acceso. Si no se encuentra ninguna, se comprueba hosts.deny; si se encuentra una regla coincidente, se deniega el acceso. Observa que el acceso puede concederse de dos maneras distintas: por coincidir en el primero o por no coincidir en el segundo; ten esto en cuenta para

no encontrarte con accesos no permitidos.

Veamos en un ejemplo; imaginemos que tenemos configurado el servidor FTP para ejecutarse a través de inetd (ya veremos cómo hacerlo en un capítulo posterior), con una línea como vimos anteriormente; si queremos que sólo accedan a él las máquinas de nuestra red excepto 192.168.0.13 tendremos que poner:

```
# contenido de /etc/host.allow
in.ftpd:    LOCAL, 192.168.0 EXCEPT 192.168.0.13
```

```
# contenido de /etc/host.deny
ALL:  ALL
```

No hay que olvidar esto último, ya que si no estaríamos permitiendo el acceso desde fuera de nuestra red.

Existen dos tipos de políticas de seguridad frecuentemente utilizadas: una de casi todo abierto y otra de casi todo cerrado. La primera consiste en permitir todo lo que no esté expresamente denegado; se implementa dejando vacío hosts.allow y denegando el servicio concreto en hosts.deny. La segunda por el contrario consiste en denegar todo lo que no esté permitido y se establece como en el ejemplo anterior, poniendo los accesos permitidos en el hosts.allow y denegando todo en el hosts.deny; esta última es en general la más adecuada.

Es posible que te tropieces alguna vez con un servicio especial llamado “portmap”; este servicio se encarga de encontrar un puerto disponible para servicios que no utilizan un número de puerto fijo sino que se acceden mediante un nombre; estos servicios se denominan de forma genérica RPC (Remote Procedure Call, o llamada a procedimiento remoto); en este caso es necesario indicar el origen mediante una IP, no es posible poner un nombre. Para protegerse ante posibles conexiones desde máquinas haciéndose pasar por otras se puede usar un nombre de cliente denominado PARANOID, que coincide con todas las máquinas cuyo nombre no coincida con su IP.

Ejercicios

- 1) Calcular exactamente el número de redes y máquinas disponibles en los tipos de redes A, B y C.
- 2) Escribe los puertos y el protocolo que utilizan los siguientes servicios: POP3,

SMTP, HTTP, FTP.

3) Suponiendo que existe la siguiente línea en /etc/inetd.conf:

```
pop3      stream  tcp  nowait      root  /usr/sbin/tcpd
/usr/sbin/pop3d
imap2     stream  tcp  nowait      root  /usr/sbin/tcpd
/usr/sbin/imapd
```

configura los TCP wrappers para que el primero no sea utilizado por la red 192.168.10.0, y el segundo pueda ser utilizado por los clientes 192.168.10.14 y 192.168.10.15; la política de seguridad debe ser del tipo “casi todo abierto”.

4) Repite el ejercicio anterior con una política de “casi todo cerrado”.

Soluciones

1) En las expresiones siguientes se han sustituidos por “x” los valores que van cambiando en cada caso:

Tipo A: 0.0.0.0 a 127.255.255.255, máscara 255.0.0.0

Primera red: 0.x.x.x, última: 127.x.x.x -> 128 redes, pero 0.0.0.0 y 127.x.x.x están reservadas = 126 redes

Máquinas: de x.0.0.0 a x.255.255.255, pero hay que restar 2 (direcciones de red y difusión) -> $256 \times 256 \times 256 = 16777216$, menos 2 = 16777214 máquinas.

Tipo B: 128.0.0.0 a 191.255.255.255, máscara 255.255.0.0

Primera red: 128.0.x.x; última red: 191.255.x.x -> $(191-128) \times 256 = 16128$

Máquinas: x.x.0.0 a x.x.255.255, a las que hay que restar 2 -> $(256 \times 256) - 2 = 65534$ máquinas.

Tipo C: 192.0.0.0 a 223.255.255.255, máscara 255.255.255.0

Primera red: 192.0.0.x; última: 223.255.255.x -> $(223-192) \times 256 \times 256 = 2031616$ redes.

Máquinas: x.x.x.0 a x.x.x.255, menos 2 -> $256 - 2 = 254$ máquinas.

2) Los puertos y protocolos son (consultar /etc/services): POP3 = 110 (tcp y udp), SMTP = 25 (tcp), HTTP o WWW = 80 (tcp y udp), FTP = 21 (tcp). Debes ir familiarizándote con ellos porque los usaremos más adelante.

3) Política “casi todo abierto”: hosts.allow vacío; host.deny con las líneas:

pop3: 192.168.10.0
imap2: ALL EXCEPT 192.168.10.14,192.168.10.15

4) Política “casi todo cerrado”: host.deny con “ALL: ALL”; host.allow:
pop3: ALL EXCEPT 192.168.10.0
imap2: 192.168.10.14,192.168.10.15

