Laboratorio Sesión 07: La red bajo Linux

En esta sesión vamos a trabajar con la red bajo Linux. Empezaremos por la configuración de la red a diferentes niveles y continuaremos por la instalación y uso de los servicios de red más básicos: `ftp`, `ssh` y páginas web.

7.1. Configuración de la red en Linux

Primero de todo vamos a saber ver que configuraciones de red tenemos en el sistema en este momento. Para ello tenemos una instrucción básica:

```
ifconfig
```

Si tecleáis esta orden os aparecerán todas las conexiones de red que tengáis en el sistema en ese momento. En vuestro ordenador aparecerán al menos dos conexiones, y siempre, aunque no tengáis red, al menos una. Linux es un sistema operativo pensado para trabajar en red, así que siempre lo hará. Es decir, un ordenador con Linux es una red de un solo ordenador. Para ello existe siempre la conexión de Loopback:

```
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:10 errors:0 dropped:0 overruns:0 frame:0
          TX packets:10 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:500 (500.0 b)  TX bytes:500 (500.0 b)
```

Esta conexión permite que, por ejemplo, un solo ordenador pueda ser cliente y servidor de páginas web a la vez, o que podamos probar configuraciones contra el propio ordenador que estamos configurando.

Además de esta conexión, `ifconfig` listará el resto de conexiones que tengamos organizadas según los siguientes nombres:

- `ethx`: tarjetas ethernet.
- `pppx`: conexiones serie (módems).
- `slx`: conexiones SLIP (conexiones serie directas).
- `fddix`: conexiones con dispositivos FDDI (fibra óptica).
- `wlanx`: conexiones inalámbricas (wifi).

Donde la x es un número que empieza en 0 y aumenta a medida que añadimos más conexiones del mismo tipo. También es interesante el comando `iwconfig` que permite gestionar las conexiones inalámbricas, pero no tenéis de estas últimas en vuestro ordenador de prácticas.

Pregunta 7.1: *¿Qué tipo de conexiones externas tenéis en vuestro ordenador de prácticas? ¿Cuáles son sus nombres y tipos? ¿Cuál es la que realmente os da conexión a internet (es la única que tiene dirección IP)?*

Lo primero que vamos a ver es como se puede apagar la red y volver a encender. Podéis apagar una conexión tecleando:

```
ifconfig nombreconexión down
```

Y volver a encenderla mediante:

```
ifconfig nombreconexión up
```

Podéis probar a hacerlo con la conexión externa de vuestro ordenador. Apagadla, comprobad mediante `ifconfig` que ya no está activada (o intentad conectaros a una página web externa) y volved a encenderla. Volved a comprobar que está activada mediante `ifconfig`.

En este momento pueden pasar dos cosas: (a) si tenéis activado el programa "NetworkManager" (por defecto) el sistema se conectará solo a la red después de hacer el `ifconfig nombreconexión up`; (b) "NetworkManager" no os conecta automáticamente y, a pesar de tener conexión, en este momento no tendréis salida a internet. Lo más probable es que estéis en el primer caso (el "NetworkManager" es el programa que gestiona la red en el entorno gráfico, podéis acceder a él a través del icono de conexión que tenéis arriba a la derecha).

Para continuar y aseguraros de tener que hacerlo todo a mano (y poder ver como va) abrid el "NetworkManager" (el icono con la red) y desactivad para la conexión que os da salida a Internet la opción de conectar automáticamente (id a la opción `Edit Connections`, y desmarcad `Automatically connect to this network when it is available`). A continuación volved a desactivar la conexión y a activarla. Ahora estaréis en el caso "b" de los dos que acabamos de comentar. Probadlo con cualquier navegador. ¿A que se debe que no funcione la red si tenéis la conexión activada? Básicamente a que al apagarla habéis perdido la configuración de la red, y así, el sistema no sabe hacia donde enviar los datos. Para solucionar esto existen varios caminos. El más fácil y habitual hoy en día es usar un sistema de configuración de red automático llamado `DHCP`. Como en el aula tenemos un servidor `DHCP`, basta con enviarle una petición:

```
dhclient nombreconexion
```

Esta orden no necesita parámetros porque se configura en el fichero `dhclient.conf`. Si todo ha ido bien, ahora la red volverá a funcionar de forma adecuada.

Pregunta 7.2: *¿Donde está ubicado el fichero* `dhclient.conf`*? ¿Que configuraciones pide al servidor* `DHCP`*?*

¿Cómo se podría hacer en caso de que no dispusiéramos del sistema `DHCP`? Bien, en ese caso deberíamos realizar la configuración a mano de 3 conjuntos de datos principales:

- La dirección de nuestro ordenador y de nuestra red.
- La dirección de los servidores de nombres a los que nos queramos conectar.
- La ruta hacia la que debemos enviar nuestros paquetes para que salgan de la red (la dirección del "gateway").

La dirección de nuestro ordenador se puede configurar directamente mediante el propio comando `ifconfig`:

```
ifconfig CONEXION IP netmask MASCARA broadcast @BROADCAST up
```

Por ejemplo (ojo, la IP es un ejemplo, para configurar la vuestra consultad la que teníais antes de apagar la red porque sino el servidor no os encontrará):

```
ifconfig eth2 10.10.41.113 netmask 255.255.255.0 broadcast 10.10.41.255 up
```

Pregunta 7.3: *¿Qué dirección IP tiene vuestro ordenador? ¿Cual es la dirección de "broadcast"? ¿Para qué sirve?*

A continuación hay que incluir en el fichero `/etc/resolv.conf` la dirección de los servidores de nombres (`DNS`).

Pregunta 7.4: *¿Qué hace un servidor DNS? ¿Qué significan cada una de las líneas que tenéis en el fichero resolv.conf?*

Finalmente, para poder acabar de configurar la red a mano deberéis establecer la ruta por defecto que tienen que seguir los paquetes. Dicha ruta se establece mediante el comando `route`.

Si lo utilizáis sin ningún parámetro podréis ver la ruta de salida por defecto que tenéis en vuestro sistema. Podéis probar ahora a desactivar la conexión de red (`ifconfig eth0 down`) y a volver a activarla, pero ahora a mano:

```
ifconfig CONEXION IP netmask MASCARA broadcast @BROADCAST up
route add default gw IPdelGATEWAY eth2
```

Fijaros que no hace falta volver a entrar los DNS ya que el fichero `resolv.conf` no se pierde, así que no hay que tocarlo. Los comandos anteriores funcionarán sin necesidad de usar el comando `dhclient`. Comprobadlo.

Como siempre pasa, además, hay herramientas de mayor nivel que realizan esta tarea. En nuestro sistema, como ya hemos comentado, tenemos una GUI llamada "NetworkManager" que se encarga de todo este sistema de forma automática. Esta GUI lee el fichero de configuración `interfaces` (que contiene la información sobre como configurar las tarjetas de red en arranque). A continuación hace un bucle para cada dispositivo de red del sistema que no se encuentra en el fichero anterior y le asigna un comportamiento por defecto (básicamente conectarse primero por cable, luego a redes inalámbricas conocidas y, finalmente, a donde se pueda).

Pregunta 7.5: *¿Qué directorio de `/etc/` almacena la información que gestiona el "NetworkManager"?*

Probad ahora a modificar alguno de los interfaces de red mediante la GUI. Podéis comprobar que el fichero se modifica. Probad por ejemplo a añadir una IP y un "GateWay" fijos a un interfaz y buscad los datos introducidos en los ficheros del sistema.

Pregunta 7.6: *¿Qué ficheros de configuración se han modificado? ¿cómo?*

7.2. La red, los servidores y los clientes

Una vez que tenemos la red funcionando ya podemos hacer cualquier cosa sobre ella. Básicamente este "cualquier cosa" significa dos cosas: o ser clientes o ser servidores.

Un cliente es un sistema que pide un servicio a otro sistema distinto, mientras que un servidor es el sistema que proporciona el servicio. Una vez que la red está instalada, ser cliente o servidor simplemente es un problema de disponer del programa que realiza dicha tarea. A diferencia de otros sistemas operativos (no diré nombres pero todos sabemos cuales ;-) Linux no esconde este funcionamiento, de forma que una vez funciona la red, tan solo se trata de activar los programas cuyos servicios nos interesan.

El uso de los programas clientes suele ser muy sencillo y todos lo habréis llevado a cabo. Usar el cliente de páginas web es simplemente instalarse el Firefox, o el Opera, o el Konqueror, o el... y usarlo. Los servidores, sin embargo son menos conocidos así que en esta práctica nos centraremos en ellos.

Para activar un servidor basta con activar el programa servidor. Los programas servidores en Linux se suelen conocer como "daemons" (demonios) y se caracterizan por ser programas que se ejecutan en segundo plano esperando una petición. Cuando les llega una petición los "daemons" se activan, realizan la tarea que tienen programada (habitualmente proporcionar el servicio) y vuelven al estado inactivo.

Es decir, un servidor es un programa que siempre está esperando una petición. ¿Y como le llega esa petición? Pues las peticiones se canalizan a través de lo que se denominan puertos. Cuando un paquete de datos llega a la tarjeta de red de un ordenador, dicho paquete contiene además el número de puerto al que va dirigido (sería como si dijéramos el piso dentro del edificio). La tarjeta de red envía los datos al puerto (en realidad el sistema operativo hace de puente) y si dicho puerto tiene un programa servidor escuchándolo, el servidor se activará y se hará cargo de los datos. ¿Cómo se sabe que hay un servidor en un determinado puerto? Pues porque hay unas asignaciones estándar que todo el mundo sigue. Podéis ver una lista de ellas en el fichero `services`. Evidentemente se puede poner un servidor en un puerto que no corresponda y configurar a los clientes de forma acorde, pero como medida de seguridad no suele funcionar y si que provoca bastantes líos a los clientes.

7.3. El servidor SSH

El primer servidor que vamos a ver es el servidor de conexiones SSH. SSH es un sistema de conexión segura (mediante encriptación) que permite abrir un terminal en la máquina servidor, de forma que podemos trabajar con ella a distancia. Además, como es un sistema texto, es rápido aunque solo dispongamos de una conexión por módem.

El programa servidor de ssh se llama sshd (Daemon SSH :-), pero se ejecuta siempre a través de un script que se encuentra en /etc/init.d/ y que en este caso se llama también ssh. En este directorio (/etc/init.d/) hay scripts para activar muchos demonios interesantes de los que solo vamos a poder ver unos pocos :-(.

Todos los scrips de /etc/init.d/ suelen tener las mismas opciones: start, stop, reload y restart. Podéis fijaros en el script y ver que se hace en cada caso. Si queréis activar el servidor basta con que tecleéis: /etc/init.d/ssh start (OJO, fijaros que si tecleáis ssh a secas lo que estaréis ejecutando es el programa cliente de ssh) o, mediante el sistema nuevo service ssh start.

Pregunta 7.7: *¿Tenéis un script* ssh *en el directorio* /etc/init.d/ *de vuestro sistema? ¿Qué paquete lo contiene si los buscáis mediante* apt-cache*? Instaladlo*

Fijaros que al instalar el paquete os arranca automáticamente el servidor. Poneros de acuerdo con los del ordenador de al lado (o de otro cercano) y probad a conectaros a su ordenador y ejecutar eject. Podéis activar y parar el servidor de forma que veréis que efectivamente a veces os podéis conectar y a veces no.

Evidentemente, el programa servidor tiene sus propias opciones de configuración. El fichero que contiene la configuración del servidor ssh es /etc/ssh/sshd_config. Echadle un vistazo.

Pregunta 7.8: *¿Que puerto escucha por defecto el servidor* ssh*?*

Finalmente, haceros notar, que con ssh trabajáis en la máquina remota. Por lo tanto, abrir un servidor ssh implica que alguien se puede conectar como root a vuestra máquina y hacer todo lo que puede hacer el root... En el tema de los servidores, la regla es que se han de activar los mínimos necesarios, cada uno es un posible fallo de seguridad.

Pregunta 7.9: *¿La configuración por defecto permite conectarse como usuario* root*? ¿Cómo evitaríamos que el usuario* root *se pudiera conectar por* ssh*?*

NOTA: No lo menciono para no repetirme constantemente, pero si alguna vez tenéis necesidad de configurar un servidor ssh, lo primero que deberíais leer es: info ssh (sobre el programa cliente), info sshd (sobre el programa servidor) e info sshd_config (sobre las opciones).

Bien, una vez que sabemos arrancar el servidor ssh, ¿cómo se puede conseguir que arranque siempre con el ordenador sin tener que arrancarlo cada vez? Cuando Linux arranca entra en lo que se denomina un nivel de ejecución (runlevel). Aunque esto en realidad es configurable y puede variar, en general todos los *nix tienen 7 niveles de ejecución:

- 0 Parada: Sirve para terminar. Se invoca con shutdown -h, halt o poweroff (o mediante la ventana).
- 1 Monousuario: Típicamente para tareas administrativas. No suele admitir ventanas.
- Niveles 2 a 5: Dependiendo del sistema, diferentes niveles de multiusuario, con mas o menos servicios.
- 6 Reinicio: Reinicia el sistema. Se invoca con shutdown -r o reboot.

Estos niveles de ejecución tienen cada uno diferentes servicios activos (o servidores o demonios). El nivel de ejecución por defecto se especifica en el fichero /etc/init/rc-sysinit.conf. Cuando se pasa de un nivel de ejecución a otro (con el comando init) o bien en el arranque,

el sistema decide que servicios activar con el contenido de los directorios `/etc/rcN.d/` donde `N` es el numero de "runlevel". Estos subdirectorios contienen cada uno unos scripts que se llaman: `KxxNOMBRE` o `SxxNOMBRE` donde `xx` es el numero de orden en el que se ejecutan. Los llamados `KxxNOMBRE` matan (Kill) el servicio "NOMBRE", mientras que los `SxxNOMBRE` lo empiezan. Así, añadiendo estos scripts al directorio adecuado podemos determinar exactamente que servicios tener activos en cada momento de una forma fácil y apagar un servidor rápidamente para pasar a mantenimiento. Aparte de configurar estos ficheros manualmente, todas las distribuciones tienen herramientas gráficas que permiten modificarlos de una forma más cómoda, o con sistemas texto como el programa `update-rc.d`, por ejemplo:

```
update-rc.d xdm enable 2 3
update-rc.d xdm disable 4 5
```

La línea anterior indica que el script `xdm` (el cual debe estar en `/etc/init.d`) debe ser ejecutado al entrar a los niveles de ejecución 2 y 3 y desactivado en los niveles 4 y 5. Fijaros además que todos los scripts de los directorios de arranque son enlaces a scripts de `/etc/init.d`.

Pregunta 7.10: *¿Cuál es el nivel de ejecución por defecto de Ubuntu?*

Pregunta 7.11: *¿Con qué orden evitaríais que el servidor* `ssh` *se arrancara cada vez que se arranca el sistema? ¿Qué fichero aparece en el directorio de inicio del nivel de ejecución por defecto al ejecutar la orden anterior? ¿Qué contiene dicho fichero?*

7.4. El servidor `telnet`

`Telnet` es un sistema que hace lo mismo que `ssh` de forma no segura. No lo uséis. Punto. Si lo encontráis en un servidor a vuestro cargo desactivadlo. Si no lo hacéis, no digáis que no os lo advertí.

7.5. El servidor `ftp`

`FTP` (File Transfer Protocol) es un sistema de intercambio de ficheros. Existen múltiples clientes `ftp` (varios de ellos gráficos y agradables), y también varios programas servidores: `ftpd`, `vsftpd`, `proftpd`... (algoFTP Daemon... supongo que vais cogiendo la idea).

El sistema que tenéis instalado es un sistema pensado para escritorio y por tanto no trae por defecto un servidor `ftp`. Así pues, lo primero que deberéis hacer es bajaros uno. Uno particularmente fiable para instalar en modo "standalone" (más sencillo) es `vsftp` (Very Secure FTP) así que os recomiendo que instaléis este, aunque si buscáis en el sistema de paquetes veréis que hay muchas otras opciones. Con todos se acaba pudiendo hacer prácticamente lo mismo aunque en algunos cuesta un poco más que en otros. Otra opción altamente recomendable es `proftp`.

Pregunta 7.12: *¿Cuántos servidores* `FTP` *diferentes podéis instalar con las fuentes de software estándar?*

Algunos ficheros interesantes del servidor `FTP`:

- `etc/ftpusers`: lista de usuarios que NO se van a poder conectar. Fijaros que por defecto no nos deja conectarnos como `root`. Esto es por motivos de seguridad.
- `/var/log/vsftpd`: log de transferencias.
- `.message`: si en algún directorio está este fichero, `ftp` lo mostrará cuando se entre en el.

Pregunta 7.13: *¿Cuál es el fichero de configuración de* `vsftpd`*? ¿Dónde está ubicado? ¿Queda activado* `vsftp` *al instalarlo? ¿Cómo se desactiva? ¿Se activará al arrancar de nuevo?*

Probad ahora a crear un nuevo usuario (con adduser por ejemplo), rearrancar el servidor `ftp`, y a conectaros. A continuación probad con `anonymous`.

Pregunta 7.14: *¿Qué mensaje os da en cada caso? ¿Cómo se puede activar la opción de que el usuario* `anonymous` *se pueda conectar por* `ftp`*? ¿Y la de que pueda escribir ficheros además de leerlos? (Recordad recargar el servidor después de modificar las configuraciones para que estas tengan efecto).*

En cualquier caso recordad que cada servidor tendrá su propio sistema de configuración y la forma concreta de realizar ciertas tareas puede variar aunque las posibilidades son prácticamente las mismas en todos los casos.

Una última opción interesante de los servidores `ftp` es hacer `chroot` a los usuarios. La mejor forma de que entendáis para que sirve es que la activéis en el fichero de configuración (probablemente tendréis que añadir la opción `allow_writeable_chroot=YES` al fichero de configuración). A continuación recargad el servidor (recordad que sino, no le afectarán los cambios) y conectaros de nuevo como el usuario `alumne`.

Pregunta 7.15: *¿En qué directorio habéis aparecido? ¿Qué pasa si ahora ejecutáis* `cd ..`*? ¿Qué significa hacer* `chroot`*?*

7.6. Servicio de páginas WEB

El servidor de paginas web mayoritario en Internet es, en todos los ámbitos, el servidor Apache. El servidor Apache funciona igual que el servidor `ssh`, instalándolo y arrancando el script que se encuentra en `/etc/init.d/`.

Pregunta 7.16: *¿Qué orden se debe ejecutar para instalar el servidor Apache?*

Aunque el servidor Apache tiene muchísimas opciones y posibilidades, simplemente arrancándolo ya podemos acceder con cualquier navegador a la página de entrada (`http://localhost/`) o a otro ordenador que se encuentre al lado. Probadlo.

Las páginas web que aparecen por defecto son simplemente ficheros de texto que se encuentran en `/var/www`. El servidor carga por defecto aquellos ficheros que se llaman `index.html` (y otros nombres predeterminados, podéis mirarlos en el fichero de configuración).

Pregunta 7.17: *¿Cómo se puede crear una página de inicio que simplemente diga "Hola" en el servidor? Escribid la lista de órdenes para hacerlo.*

Otra opción interesante es que cada usuario pueda crearse su propia página de internet. Esto es muy fácil de configurar. A partir de la versión 2 del servidor Apache (la que estáis usando) cada nueva funcionalidad es un nuevo módulo que puede haber que integrar en el sistema. Para hacerlo se usa la orden `a2enmod`. En concreto, si queréis integrar las páginas de los usuarios, deberéis escribir `a2enmod userdir`. Podéis ver que módulos tenéis disponibles listando el directorio `/etc/apache2/mods-available` y los activados en el directorio `/etc/apache2/mods-enabled`. Fijaros que el fichero de configuración asociado implica que cualquier usuario puede tener una zona propia poniéndola bajo un directorio llamado `public_html` en su propio `home`. A está página se accederá mediante `http://nombredelservidor/~nombredeusuario`.

Pregunta 7.18: *¿Qué ordenes hay que teclear como* `root` *para que cualquier usuario tenga una zona web personal? ¿Qué deberíamos modificar para que el directorio de la página web personal se llamara "WebPersonal"?*

Pregunta 7.19: *¿Cómo puede el usuario* `alumne` *crear una página web propia que simplemente diga "Tonto el que lo lea"? Hacedlo, probadla y escribid la lista de órdenes que ha tenido que teclear el usuario para que funcione.*

Antes de seguir parad el servidor apache.

7.7. Seguridad en redes

Uno de los elementos más importantes a la hora de mantener una red de área local es la seguridad. Una red insegura puede ser fuente de muchos problemas y dejar de funcionar, hacerlo mal o, incluso, convertirnos en víctimas de una estafa.

Para mantener la seguridad, además de preocuparnos de configurar la red de la forma más segura posible y de no ejecutar ningún servidor innecesario, es bueno tener algún tipo de herramienta que nos permita mejorar el control que tenemos sobre nuestro sistema.

Una herramienta muy interesante para esto es, por ejemplo, `Lynis` (hay más), un buscador de Rootkits en nuestro sistema, es decir, de agujeros de seguridad conocidos y explotables (solo que en vez de buscarlos para explotarlos, los busca para poder cerrarlos). Es muy fácil de usar y permite eliminar los fallos más graves del sistema fácilmente.

Conectaros a la página web (`https://www.rootkit.nl/projects`) y descargad el programa. Podéis intentar instalarlo a mano o utilizar el `apt-get` (aunque la versión de la web suele estar más actualizada). En la misma web veréis que hay más formas de instalarlo.

Pregunta 7.20: *¿Qué opciones tenéis para instalar el programa? ¿Qué pasos habéis seguido vosotros para instalar el programa?*

A continuación podéis probar a ejecutarlo (`./lynis -c`) para ver que tal es la seguridad del sistema instalado en el aula por defecto.

Pregunta 7.21: *¿Qué aviso nos da el programa si lo ejecutamos sin ser superusuarios? ¿Y si lo intentamos ejecutar como superusuario? Explicad la razón (según como hayáis instalado el programa los avisos pueden cambiar o no aparecer).*

Arreglad el problema para poder ejecutar el programa como superusuarios (si os daba alguno) y ejecutad nuevamente. Como veis, el programa es muy claro sobre lo que está bien, lo que puede estar mal y lo que, definitivamente, debe cambiarse. Fijaros que además el programa acaba con una lista de sugerencias de acciones concretas a realizar.

Ahora sería el momento de corregir los posibles errores que tenga el sistema, actualizando los programas viejos o corrigiendo directamente problemas de configuración, como los que tiene el servidor `ssh` (si no tenéis un servidor `ssh` funcionando instaladlo, arrancadlo y volved a ejecutar `Lynis`).

Pregunta 7.22: *¿Qué errores reporta con respecto al sistema* `grub`*? ¿Qué* `Hardening Index` *le da a vuestro sistema?*

Probad ahora a corregir al menos un error (basta con que sigáis las sugerencias, si cambiáis alguna configuración recordad recargar el servidor para que la actualice) y volved a ejecutar `lynis`.

Pregunta 7.23: *¿Qué habéis corregido? ¿Cómo? ¿Cuál es el nuevo* `Hardening Index` *de vuestro sistema?*

Finalmente podéis cotillear un poco la web de la empresa para ver la idea de negocio de una herramienta de este tipo.

Pregunta 7.24: *¿Qué otros productos ofrece la empresa? ¿Son de software libre? ¿Cuál es su modelo de negocio?*

Como podéis ver, la distribución `Ubuntu` es bastante segura de por si. Prácticamente basta con tenerla actualizada para que sea segura. De todas formas, si instaláis un servidor web `Apache` considerad utilizar la herramienta `mod_security`. Podéis buscar la página web y echarle un vistazo.