Colección Certificaciones

Preparación para la certificación LPIC-2

LINUX

32 prácticas
110 preguntas-respuestas

2ª edición

Exámenes LPI 201 y LPI 202

GRATIS:
UN EXAMEN EN BLANCO en línea
con respuestas comentadas y detalladas

eni ediciones

Descarga
www.ediciones-eni.com

Sébastien BOBILLIER

LINUX

Preparación para la certificación LPIC-2 (exámenes LPI 201 y LPI 202) - 2ª edición

Los exámenes **LPI 201 y LPI 202** son los dos exámenes que permiten obtener la **certificación LPIC-2** "Advanced Level Linux Professional". Este programa de certificación del Linux Professional Institute está cada vez más**reconocido por las empresas de selección**, que ven en esta certificación un requisito para la contratación o el acceso a un puesto de administrador.

Los exámenes LPI 201 y 202 demuestran a los profesionales que usted domina la administración avanzada de un sistema **Linux de cualquier distribución**: califican las competencias prácticas en términos de administración de redes de pequeño o mediano tamaño (administración de servicios de red comunes, gestión de la seguridad de red y de las comunicaciones...).

Para ayudarle a preparar eficazmente esta certificación, este libro cubre**todos los objetivos oficiales de la última versión del examen** (implantada en julio de 2012) tanto desde un punto de vista teórico como práctico. Ha sido originalmente redactado por un profesional formador y consultor, certificado en Linux. De este modo, el saber pedagógico y técnico del autor conducen a una aproximación clara y visual, de un nivel técnico muy alto.

Capítulo a capítulo, podrá **validar sus conocimientos teóricos**, con la ayuda de múltiples **preguntas-respuestas (110 en total)** que ponen de relieve tanto los elementos fundamentales como las características específicas de los conceptos tratados.

Cada capítulo finaliza con unos **trabajos prácticos (32 en total)** en los que puede medir su autonomía. Estas operaciones concretas, más allá incluso de los objetivos marcados para el examen, le permitirán construir una primera experiencia significativa y adquirir verdaderas competencias técnicas en situaciones reales.

A este dominio de práctica y de conocimientos, se añade la preparación específica para la certificación: podrá acceder **gratuitamente a 1 examen en blanco en línea**, para que pueda practicar en condiciones parecidas a las de la prueba.

Los capítulos del libro:
Prólogo – Introducción – Administración del almacenamiento – Arranque del sistema – Administración de la red local – Autentificación de usuarios – Compartición de archivos – Resolución de nombres DNS – Servidor web Apache – Correo electrónico – Protección de redes – Asegurar las comunicaciones – Compilación de aplicaciones y del kernel Linux

---

Sébastien BOBILLIER

Después de haber sido Administrador de Sistemas y Redes, **Sébastien Bobillier** evoluciona durante muchos años en el mundo de la formación. Hoy en día, es Consultor Formador en Global Knowledge, y se ha convertido en un especialista de sistemas Linux, que acompaña regularmente a los candidatos a la certificación LPI. Este libro es el fruto de toda su experiencia en esta materia.

Preparación para la Certificación LINUX LPIC-2 exámenes LPI 201 y LPI 202

Los exámenes **LPI 201 y LPI 202** son los dos exámenes que permiten obtener la **certificación LPIC-2** "Advanced Level Linux Professional". Este programa de certificación del Linux Professional Institute está cada vez más **reconocido por las empresas de selección** que ven en esta certificación un requerimiento para la contratación o para el acceso a un puesto de administrador.

Los exámenes LPI 201 y 202 demuestran a los profesionales que usted domina la administración avanzada de un sistema **Linux de cualquier distribución**: califican las competencias prácticas en términos de administración de redes de pequeño o mediano tamaño (administración de servicios de red comunes, gestión de la seguridad y de las comunicaciones...).

Para ayudarle a preparar eficazmente esta certificación, este libro cubre los objetivos oficiales cuya lista se proporciona en el anexo. Se divide en 12 capítulos organizados de la siguiente manera:

- Una definición de los objetivos que se alcanzarán: permite exponer de forma precisa las competencias adquiridas cuando finalice el capítulo.
- Una parte de **conocimientos teóricos**: permite definir los términos y conceptos tratados y esquematizar en forma de hilo conductor los distintos puntos que se deben asimilar.
- Una parte de **comprobación de los conocimientos adquiridos** escrita en forma de preguntas/respuestas (110 en total). Estas preguntas se centran tanto en los fundamentos como en las características específicas de los conceptos tratados. La parte **respuestas** describe la respuesta correcta para cada una de las preguntas realizadas.
- Los **trabajos prácticos**: permiten ilustrar de forma precisa algunas partes del capítulo y también proporcionan los medios para medir su autonomía. Estas operaciones concretas, más allá de los propios objetivos fijados para el examen, le permitirán obtener una primera experiencia significativa y adquirir verdaderas habilidades técnicas como si se tratase de situaciones reales.

En la preparación específica para el examen, puede acceder **gratuitamente a 1 examen en blanco** en línea en la dirección http://www.edieni.com para entrenarse en condiciones muy parecidas a las de la prueba. En este sitio web, cada pregunta se ha escrito inspirándose en la certificación y, para cada una de ellas, las respuestas están lo suficientemente comentadas para controlar e identificar sus últimas lagunas.

La certificación LPI

1. Interés de la certificación

El universo Open Source rebosa de personas con habilidades dispares y con fundamentos más o menos sólidos. Hay gran cantidad de gurús por Internet y los trabajos de algunos aficionados son exagerados, quizá superiores a los que realizan los empleados expertos en la materia.

Las empresas podrían estar encantadas con esta masa de conocimiento disponible y quizás incluso contratar con menor coste a estos usuarios apasionados. El problema es que a menudo estos conocimientos se adquieren de forma autodidacta, sin seguir ninguna metodología y frecuentemente fuera de cualquier marco profesional, lo que impide a los candidatos aportar documentación de sus servicios en el ámbito empresarial. Por otro lado, el aprendizaje autodidacta no se realiza por amor al software, y los usuarios aficionados a menudo tienen una visión fragmentada y sesgada de la materia, centrada en sus intereses personales.

Los programas de certificación tienen por objetivo validar unas habilidades, independientemente del curso universitario o escolar. Sirven para validar un nivel concreto y, en lo concerniente a la certificación LPI, para verificar que el candidato tiene una visión transversal de la materia, sin carencias manifiestas en ninguno de los temas tratados.

2. La certificación LPI en pocas palabras

El programa de certificación LPI es el principal programa de certificación Linux multidistribución e independiente de cualquier editor. Fue creado por una comunidad de profesionales y académicos del mundo Linux. Los exámenes se han creado para detectar las carencias en cualquiera de los temas tratados. Puede presentarse en cualquiera de los centros de examen autorizados Pearson VUE o Prometric.

El programa de certificación LPI está reconocido por una gran cantidad de fabricantes y profesionales del sector como IBM, Novell, Intel o HP. Además, es un requisito para otros programas de certificación como Ubuntu o suministrado por otros editores como Suse. Hasta la fecha, hay más de 85.000 certificados LPI en el mundo, con más de 250.000 exámenes realizados.

3. El programa de la certificación LPI

a. Nivel 1

El nivel 1 de la certificación LPI se obtiene aprobando los dos exámenes LPI 101 y 102. Califica un conocimiento básico (lo que no significa que sea fácil) de los sistemas Linux, de los comandos básicos y del shell. Estas habilidades se pueden considerar como un requisito en el progreso serio en la administración de sistemas Linux. La adquisición de las competencias correspondientes a la certificación LPI Nivel 1 no conduce a la total autonomía en la materia, pero sí a un buen nivel de confort en la ejecución de las tareas enmarcadas.

b. Nivel 2

El nivel 2 de la certificación LPI se obtiene aprobando los dos exámenes LPI 201 y 202. Califica un conocimiento práctico de la administración de redes de tamaño pequeño o mediano y de la administración de los servicios de red corrientes. También evalúa la gestión de la seguridad de la red y de las comunicaciones. Un administrador de sistemas certificado con LPI nivel 2 es autónomo administrando su red y sus equipos.

c. Nivel 3

El nivel 3 de la certificación LPI se obtiene aprobando el examen LPI 301, al que se le pueden añadir cinco exámenes de especialización numerados del 302 al 306 que tratan los entornos mixtos, la seguridad, la alta disponibilidad y la virtualización, las tecnologías de Internet y, por último, la mensajería. La certificación LPI nivel 3 se presenta como el máximo nivel de certificaciones Linux.

4. Presentarse al examen

Presentarse al examen de una certificación LPI deja a menudo un amargo sabor de boca. El nivel de dificultad de las preguntas parece inalcanzable. A menudo se llega a pensar que esta certificación es injusta, porque no califica realmente una habilidad sino más bien la capacidad de aprender de memoria el manual de comandos de Linux. Naturalmente, esta proeza no está al alcance de los seres humanos en la vida real, hay que buscar en otro sitio la clave del éxito.

Como en todas las preguntas de opción múltiple, en caso de duda primero debemos eliminar las respuestas extravagantes; la respuesta correcta a menudo se encuentra en una lista reducida de dos o tres alternativas. Además, hay que recordar que la certificación LPI se basa por un lado en los conocimientos teóricos y, por otro lado y sobre todo, en unas competencias prácticas adquiridas sobre los sistemas Linux. En principio un candidato se instruye, pasa unos cuantos años administrando sistemas en producción y se presenta a la certificación para demostrar que ha asimilado los conocimientos técnicos. Es evidente que no todos siguen este proceso y, para muchos, incluyendo el mercado laboral europeo, la certificación es vista en gran medida como una manera de conseguir el empleo que aportará la experiencia real. Esta ecuación aparentemente irresoluble puede resolverse, y este libro está aquí para ayudar. Simplemente, no es suficiente con haber realizado todos los ejercicios y trabajos prácticos, sino que también será necesario haberlos asimilado. Es decir, hay que entender los conceptos subyacentes, el interés de los comandos usados y ser capaz de emplearlos fuera del contexto de los ejercicios.

Si no se tiene a mano una infraestructura Linux en producción para entrenarse, un buen método es leer este libro y realizar los trabajos prácticos que se encuentran al final de cada capítulo. Al principio se sentirá tranquilo al ver que todo funciona correctamente (si se siguen escrupulosamente las instrucciones, todos los ejercicios funcionan). De hecho, es muy importante no desanimarse: la certificación LPI exige que uno se sienta muy cómodo con todos los conceptos y comandos. Una lectura desmotivada y el aprendizaje de comandos permitidos sin convicción y sin comprenderlos no bastará. Una vez tranquilizado por haber hecho bien los ejercicios, se puede volver a leer, profundizar en los puntos oscuros y animarse a probar alternativas a los ejercicios. En los trabajos prácticos se dejan abiertas muchas posibilidades, el lector es libre de realizarlas.

Hoy en día los exámenes LPI siempre están en inglés. Se recomienda tener un buen nivel de inglés técnico.

Sobre este libro

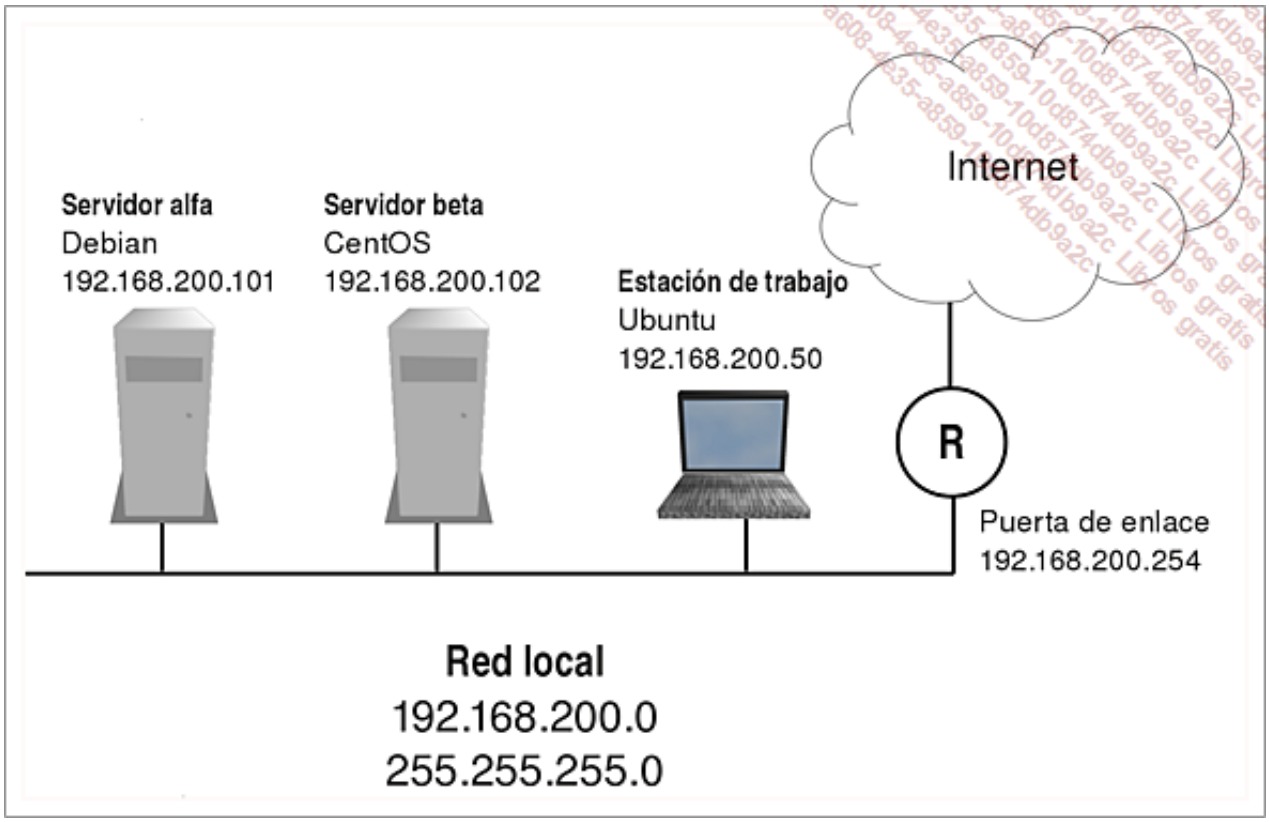
1. La información técnica

Este libro está dirigido principalmente a la preparación para la certificación LPI. Por lo tanto su contenido técnico se orienta en esta dirección. Algunos detalles funcionales o ciertos comandos expuestos han caído un poco en desuso, aunque la certificación LPI exige su conocimiento.

La certificación LPI nivel 2 califica que los candidatos dispongan de un excelente conocimiento práctico de los sistemas Linux y de los servicios de aplicación más comunes. Las preguntas a veces son con trampa, precisamente para comprobar que el candidato posee experiencia administrando sistemas en una determinada materia y que ya se ha encontrado con situaciones particulares al margen del funcionamiento diario "cuando todo va bien". De este modo, aquí encontrará las explicaciones, los conocimientos y en la medida de lo posible muchos trucos que una extensa práctica debería aportar.

Por supuesto, la información sobre los comandos y aplicaciones Linux son de dominio público y ampliamente disponibles, no sólo por el manual en línea. Las sintaxis de los comandos expuestos aquí sólo son las opciones realmente importantes: ya sea porque son de uso común en producción o porque los objetivos específicos de la certificación LPI los considera particularmente importantes. Por lo tanto el candidato puede, al menos inicialmente, centrarse en el conocimiento de lo esencial.

2. Los trabajos prácticos



Los trabajos prácticos propuestos se basan en un entorno mixto compuesto por dos servidores y una estación de trabajo Linux. El primero de los servidores tendrá instalada una distribución Debian y el otro una distribución CentOS, que tiene como ventaja su proximidad con los sistemas Red Hat, siendo mucho más fácil de obtener. La estación de trabajo se instalará con una distribución Ubuntu.

Durante un ejercicio sobre cómo instalar un gestor de arranque se utilizará puntualmente un live CD de la distribución DSL (*Damn Small Linux*).

Las máquinas tendrán como nombre de host **alfa**, para el servidor Debian, y **beta**, para el servidor CentOS. La estación de trabajo con Ubuntu se llamará **estacion**. Las direcciones IP deberán acogerse a su plan de direccionamiento y son irrelevantes para la realización de los ejercicios, aunque deberán ser coherentes. Las direcciones utilizadas para los trabajos prácticos serán 192.168.200.101 para el servidor **alfa**, 192.168.200.102 para el servidor **beta**, y una dirección cualquiera de la misma subred para la estación de trabajo. Se deberá reemplazar estas direcciones por las que se haya elegido.

El entorno de trabajo es virtualizado para permitir un montaje fácil de una maqueta realista sin tener que desplegar una cantidad considerable de hardware. Además, la virtualización tiene la ventaja de que permite realizar operaciones pesadas de almacenamiento a un menor coste. El software de virtualización elegido es VirtualBox OSE, que tiene la ventaja de estar disponible de forma gratuita y de poderse instalar tanto en equipos Windows como en equipos Linux. La adaptación a cualquier otro software de virtualización no debería presentar ningún tipo de dificultad añadida.

Si se desea trabajar en un entorno real los trabajos prácticos se pueden adaptar muy fácilmente con tres equipos, teniendo en uno de ellos dos tarjetas de red, un switch y algunos cables de red. En este caso se necesitarán algunos discos duros adicionales.

Sea cual sea el entorno elegido se necesitará acceso a Internet para la realización de la mayor parte de los ejercicios.

Preparación de los trabajos prácticos

1. Descarga del software

- El software de virtualización VirtualBox se puede descargar en la dirección siguiente:http://www.virtualbox.org/wiki/Downloads
- La imagen iso de la distribución Debian se puede descargar en la dirección siguiente:http://www.debian.org/CD/netinst. La versión "net install" de la distribución es ligera y los componentes adicionales se instalarán bajo demanda.
- La imagen iso de la distribución CentOS se puede descargar en la dirección siguiente:http://mirror.centos.org/centos/6/isos. Descargue la versión DVD.
- La imagen iso de la distribución Ubuntu se puede descargar en la dirección siguiente:http://www.ubuntu.com/desktop/get-ubuntu/download
- La imagen iso de la distribución DSL se puede descargar en la dirección siguiente:http://www.damnsmalllinux.org/download.html

Los trabajos prácticos se han realizado con la versión 4.1.12 de VirtualBox, con la versión Squeeze (6) de Debian, con la versión 6 de CentOS y con la versión Precise Pangolin (12.04) de Ubuntu. El uso de versiones distintas no debería interferir en la realización de los ejercicios.

A menudo se recomienda elegir las versiones de 32 bits de los sistemas (i386) para trabajar en un entorno virtualizado.

2. Instalación del servidor alfa

a. Elementos necesarios

- Tener el software VirtualBox instalado.
- Imagen iso de un cd Debian netinstall.

b. Creación de la máquina virtual

- Desde la interfaz de VirtualBox, haga clic en **Nueva** para iniciar el asistente de creación de máquinas virtuales.
- En la pantalla **Nombre y sistema operativo**, introduzca **alfa** en el campo **Nombre**, seleccione**Linux** como **Sistema operativo** y **Debian** como **Versión**.
- En la pantalla **Memoria**, regule el **Tamaño de memoria base** a 128 MB como mínimo. Este valor tiene que ser suficiente para una instalación sin interfaz gráfica. Si opta por instalar un servidor X deberá aumentar la memoria en consecuencia.
- En la pantalla **Unidad de disco duro**, deje los parámetros por defecto.
- En la pantalla **Tipo de archivo de unidad de disco duro**, conserve la opción por defecto.
- En la pantalla **Ubicación del archivo y tamaño**, deje la opción por defecto. El tamaño de 8 GB mostrado no se ocupará realmente en su disco duro.
- En la pantalla **Resumen**, haga clic en **Crear** y una vez más en **Crear** para finalizar la creación de la máquina virtual.

c. Configuración de la máquina virtual

- En la interfaz de VirtualBox, seleccione su máquina virtual en el panel de la izquierda.
- En el panel de la derecha, haga clic en el enlace **Almacenamiento**.
- En la pantalla **Almacenamiento/Árbol de almacenamiento**, haga clic en el cdrom (**vacío**).

- En la pantalla **Almacenamiento/Atributos**, despliegue el menú **Dispositivo CD/DVD** y elija**Seleccionar un archivo de disco virtual de CD/DVD**. En la ventana de selección de archivo elija la imagen ISO del cd virtual Debian. Acepte.
- En la interfaz de VirtualBox, seleccione su máquina virtual en el panel de la izquierda.
- En el panel de la derecha, haga clic en el enlace **Red**.
- En la ventana **Red**, modifique el valor del desplegable **Conectado a** seleccionando **Adaptador puente**. Acepte los cambios.

d. Arranque de la máquina virtual e instalación del sistema

- Desde la interfaz de VirtualBox, seleccione su máquina virtual en el panel de la izquierda.
- Haga clic en el botón **Iniciar** para arrancar la máquina virtual.
- Haga clic en la pantalla de la máquina virtual iniciada para que se capturen el ratón y el teclado.
- En el **Installer boot menu**, elija **Install**.
- En la pantalla **Choose language**, elija **Spanish**.
- En la pantalla **Seleccione su ubicación**, seleccione su país o región (por defecto es **España**).
- En la pantalla **Elija la distribución del teclado**, deje seleccionada la opción por defecto (**Español**).
- En la pantalla **Configurar la red**, borre el texto por defecto e introduzca **alfa**.
- En la pantalla **Configurar usuarios y contraseñas**, introduzca **password** como **Clave del superusuario**. Confirme la contraseña.
- En la pantalla **Configurar usuarios y contraseñas**, introduzca **usuario** como **Nombre completo para el nuevo usuario**.
- En la pantalla **Configurar usuarios y contraseñas**, introduzca **usuario** como **Nombre de usuario para la cuenta**.
- En la pantalla **Configurar usuarios y contraseñas**, introduzca **password** como **Contraseña para el nuevo usuario**. Confirme la contraseña.
- En la pantalla **Particionado de discos**, elija **Guiado - utilizar todo el disco**.
- En la pantalla siguiente **Particionado de discos**, elija el único disco mostrado.
- En la pantalla siguiente **Particionado de discos**, elija **Todos los ficheros en una partición (recomendado para novatos)**.
- En la pantalla siguiente **Particionado de discos**, elija **Finalizar el particionado y escribir los cambios en el disco**.
- En la pantalla siguiente **Particionado de discos**, valide la configuración del disco seleccionando**Sí**.
- En la pantalla **Configurar el gestor de paquetes**, elija **España** como **País de la réplica de Debian**.
- En la pantalla **Configurar el gestor de paquetes**, elija una réplica cualquiera como **Réplica de Debian**.
- En la pantalla **Configuración de popularity-contest**, seleccione **No** para rechazar el envío de estadísticas sobre el uso de los paquetes.
- En la pantalla **Selección de programas**, quite la opción **Entorno de escritorio** y deje seleccionado **Sistema estándar**.
- En la pantalla **Instalar el cargador de arranque GRUB en un disco duro**, confirme la instalación de GRUB en el registro principal de arranque.
- En la pantalla **Terminar la instalación**, seleccione **Continuar** para aceptar el final de instalación.

3. Instalación del servidor beta

a. Elementos necesarios

- Tener el software VirtualBox instalado.
- Imagen iso de un dvd CentOS.

b. Creación de la máquina virtual

- Desde la interfaz de VirtualBox, haga clic en **Nueva** para iniciar el asistente de creación de máquinas virtuales.
- En la pantalla **Nombre y sistema operativo**, introduzca **beta** en el campo **Nombre**, seleccione**Linux** como **Sistema operativo**, y **Red Hat** como **Versión**.
- En la pantalla **Memoria**, regule el **Tamaño de memoria base** a 1024 MB. Si su sistema anfitrión no puede proporcionar tanta memoria, puede disminuir este valor y optar por no instalar la interfaz gráfica.
- En la pantalla **Memoria**, regule el **Tamaño de memoria base** a al menos 768 MB. Si no puede asignar tanta memoria a su máquina virtual, puede optar por una versión más antigua de CentOS. La versión 5 se puede instalar con 256 MB de memoria base. La instalación de la versión 6 con una cantidad 26de memoria demasiado pequeña desemboca en una instalación degradada de un sistema operativo inutilizable..
- En la pantalla **Unidad de disco duro**, deje los parámetros por defecto.
- En la pantalla **Tipo de archivo de unidad de disco duro**, conserve la opción por defecto.
- En la pantalla **Almacenamiento en unidad de disco duro físico**, deje la opción por defecto.
- En la pantalla **Ubicación del archivo y tamaño**, deje la opción por defecto. El tamaño que se muestra de 8 GB no se ocupará realmente en su disco duro.
- En la pantalla **Resumen**, haga clic en **Crear** y una vez más en **Crear** para finalizar la creación de la máquina virtual.

c. Configuración de la máquina virtual

- En la interfaz de VirtualBox, seleccione su máquina virtual en el panel de la izquierda.
- En el panel de la derecha, haga clic en el enlace **Almacenamiento**.
- En la pantalla **Almacenamiento/Árbol de almacenamiento**, haga clic en el cdrom (**vacío**).
- En la pantalla **Almacenamiento/Atributos**, despliegue el menú **Dispositivo CD/DVD** y elija**Seleccionar un archivo de disco virtual de CD/DVD**. En la ventana de selección elija la imagen ISO del dvd virtual de ContOS. Acepte.
- En la interfaz de VirtualBox, seleccione su máquina virtual en el panel de la izquierda.
- En el panel de la derecha, haga clic en el enlace **Red**.
- En la ventana **Red**, modifique el valor del desplegable **Conectado a** seleccionando **Adaptador puente**. Acepte los cambios.

d. Arranque de la máquina virtual e instalación del sistema

- En la interfaz de VirtualBox, seleccione su máquina virtual en el panel de la izquierda.
- Haga clic en el botón **Iniciar** para arrancar la máquina virtual.
- En la pantalla de bienvenida, seleccione **Install or upgrade an existing system**.
- En la pantalla **Disc found**, seleccione **Skip** para evitar la comprobación del disco duro.
- En la pantalla de selección del idioma, elija **Spanish**.
- En la pantalla de selección del teclado, elija **Español**.

- Seleccione **Dispositivos de almacenamiento básicos** en la pantalla de elección del tipo de dispositivos.
- En la ventana **Advertencia del dispositivo de almacenamiento**, elija **Sí, descarte todos los datos**.
- En la pantalla siguiente, escriba **beta** como nombre del host y haga clic en **Configure la red**.
- En la ventana **Conexiones de red**, seleccione **System eth0** y haga clic en **Editar**.
- En la pestaña **Ajustes de IPv4**, elija **Manual** como método de configuración. A continuación, haga clic en **Añadir** para configurar su dirección IP.
- Configure una dirección IP en su subred conectada a Internet. En todos los ejercicios, utilizaremos la dirección 192.168.200.102/24 que deberá reemplazar por una dirección compatible con su rango de direcciones.
- Indique su puerta de enlace local y el servidor DNS de su proveedor de servicios de Internet. Acepte los cambios.
- En la pantalla de selección del huso horario, desactive **El reloj de sistema utiliza UTC** y seleccione su huso horario.
- En la pantalla de gestión de la contraseña de root, escriba **password** como contraseña. Pulse**Utilizar de todos modos** aunque sea demasiado débil.
- En la pantalla de elección del tipo de instalación, seleccione **Usar todo el espacio**. Acepte el aviso de escritura de modificaciones en el disco.
- En la pantalla de elección del tipo de instalación, elija **Desktop** y seleccione **Personalizar ahora** (en la parte de abajo de la pantalla) antes de continuar.
- En la pantalla de personalización, seleccione **Servidores/Servidores de almacenaje de red** y**Servidores de infraestructura de red**. En los paquetes opcionales de **Servidores de infraestructura de red** elija **bind-chroot**. En la sección **Desarrollo**, elija **Desarrollo de plataforma de escritorio**, **Desarrollo adicional** y **Herramientas de desarrollo**. Valide esta selección haciendo clic en **Siguiente**.
- Espere a que finalice la instalación y pulse **Reiniciar**.

e. Configuración del sistema instalado

- Después del reinicio del sistema, haga clic en **Al frente** en la pantalla de bienvenida.
- Acepte la licencia de uso.
- En la pantalla **Crear Usuario**, cree una cuenta para el usuario **usuario** con la contraseña**password**.
- En la pantalla **Fecha y Hora**, modifique si fuera necesario el reloj del sistema.
- Acepte el aviso de la imposibilidad de configurar kdump.
- Termine el asistente y abra una sesión.
- Desde un terminal conectado como root, desactive el cortafuegos tecleando **chkconfig iptables --level 2345 off**.

4. Instalación de la estación de trabajo

a. Elementos necesarios

- Tener el software VirtualBox instalado.
- Imagen iso de un cd de Ubuntu.

b. Creación de la máquina virtual

- Desde la interfaz de VirtualBox, haga clic en **Nueva** para iniciar el asistente de creación de

máquinas virtuales.

- En la pantalla **Nombre y sistema operativo**, introduzca **estacion** en el campo **Nombre**, seleccione **Linux** como **Sistema operativo**, y **Ubuntu** como **Versión**.
- En la pantalla **Memoria**, regule el **Tamaño de memoria base** a 512 MB. Si su sistema anfitrión no puede proporcionar tanta memoria puede disminuir este valor y elegir otro sistema que consuma menos recursos como xubuntu.
- En la pantalla **Unidad de disco duro**, deje los parámetros por defecto.
- En la pantalla **Tipo de archivo de unidad de disco duro**, deje los valores por defecto.
- En la pantalla **Almacenamiento en unidad de disco duro fésico**, deje la opción por defecto.
- En la pantalla **Ubicación del archivo y tamaño**, conserve las opciones por defecto. El tamaño que se muestra de 8 GB no se ocupará realmente en su disco duro.
- En la pantalla **Resumen**, haga clic en **Crear** y una vez más en **Crear** para finalizar la creación de la máquina virtual.

c. Configuración de la máquina virtual

- En la interfaz de VirtualBox, seleccione su máquina virtual en el panel de la izquierda.
- En el panel de la derecha, haga clic en el enlace **Almacenamiento**.
- En la pantalla **Almacenamiento/Árbol de almacenamiento**, haga clic en el cdrom (**vacío**).
- En la pantalla **Almacenamiento/Atributos**, despliegue el menú **Dispositivo CD/DVD** y elija**Seleccionar un archivo de disco virtual de CD/DVD**. En la ventana de selección de archivo elija la imagen ISO del cd virtual de Ubuntu. Acepte.
- En la interfaz de VirtualBox, seleccione su máquina virtual en el panel de la izquierda.
- En el panel de la derecha, haga clic en el enlace **Red**.
- En la ventana **Red**, modifique el valor del desplegable **Conectado a** seleccionando **Adaptador puente**. Acepte los cambios.

d. Arranque de la máquina virtual e instalación del sistema

- En la interfaz de VirtualBox, seleccione su máquina virtual en el panel de la izquierda.
- Haga clic en el botón **Iniciar** para arrancar la máquina virtual.
- Haga clic en la pantalla de la máquina virtual iniciada para capturar el ratón y el teclado.
- En la pantalla **Install**, seleccione **Español** en el panel de la izquierda y haga clic en **Instalar Ubuntu**.
- En la pantalla **Preparando la instalación de Ubuntu**, compruebe que cumple los requisitos y haga clic en **Continuar**. Las actualizaciones mientras se instala y el software de terceros nos son indispensables.
- En la pantalla **Tipo de instalación**, seleccione **Borrar disco e instalar Ubuntu** y haga clic en**Continuar**.
- Valide su selección pulsando **Instalar ahora**.
- En la pantalla **¿Dónde se encuentra?**, compruebe la hora y el huso horario.
- En la pantalla **Distribución del teclado**, compruebe la opción propuesta.
- En la pantalla **¿Quién es usted?**, utilice la cuenta de usuario **usuario** y la contraseña**password**. Modifique el nombre del equipo llamándole **estacion**.
- Deje que la instalación continúe y reinicie cuando el instalador se lo proponga.

e. Configuración de la dirección IP del equipo

- En la estación de trabajo Ubuntu, haga clic en el icono de conexiones de red (en la barra de

tareas superior, a la derecha). Elija **Editar las conexiones**.

- En la ventana **Conexiones de red**, haga clic en **Añadir**.
- En la ventana **Editando Conexión cableada 1**, informe el campo **Nombre de la conexión** con el valor **Fija eth0**.
- En la pestaña **Ajustes de IPv4**, elija el **Método: Manual**. Añada una dirección IP con una dirección de su rango de direcciones. Utilice su puerta de enlace predeterminada y utilice el servidor DNS de su proveedor de servicios de Internet. Valide su configuración haciendo clic en**Guardar**.
- Si fuera necesario, haga clic en el icono de gestión de red para elegir la conexión **Fija eth0**.

La cuenta root está desactivada por defecto en los sistemas Ubuntu. Todos los comandos que requieren permisos de administrador deberán ser precedidos por el comando sudo.

5. Añadir un periférico adicional a una máquina previamente creada

Las operaciones siguientes no tienen que realizarse inmediatamente. Algunos ejercicios necesitarán que se agregue hardware en algunas máquinas virtuales.

a. Agregar un disco duro (SATA)

La adición de un disco duro se realiza en la propiedades de la máquina virtual en cuestión.

b. Asociación del disco duro a la máquina virtual

- En la interfaz de VirtualBox, seleccione su máquina virtual en el panel de la izquierda.
- En el panel de la derecha, haga clic en el enlace **Almacenamiento**.
- En la pantalla **Almacenamiento/Árbol de almacenamiento**, haga clic en el botón **Agregar controlador**. Elija un controlador de tipo SATA.
- En la sección **Controlador SATA**, haga clic en el botón de agregación de disco y seleccione**Agregar disco duro**.
- En la ventana de elección del nuevo disco, seleccione **Crear nuevo disco** y acepte las elecciones por defecto.

c. Agregar una tarjeta de red

Cada máquina virtual dispone de un conjunto de cuatro tarjetas de red de las cuales sólo la primera está activa. Agregar una tarjeta de red es tan fácil como activar una de las tarjetas que ya vienen preinstaladas.

d. Activación de la tarjeta de red en la máquina virtual

- En la interfaz de VirtualBox, seleccione su máquina virtual en el panel de la izquierda.
- En el panel de la derecha, haga clic en el enlace **Red**.
- Seleccione la pestaña **Adaptador 2** y seleccione la opción **Habilitar adaptador de red**.
- Elija el modo de conexión en función de sus necesidades.

Requisitos y objetivos

1. Requisitos

Los conocimientos adquiridos con la certificación LPI nivel 1, en particular:

- ☒ Tener nociones básicas acerca de los sistemas de archivos y de las tablas de inodos.
- ☒ Conocer el particionado tradicional de discos de PC.
- ☒ Utilización básica de la utilidad fdisk.
- ☒ Conocimientos básicos de almacenamiento en cinta magnética (/dev/st*, /dev/nst* y mt).

2. Objetivos

Al finalizar del capítulo será capaz de:

- ☒ Conocer las principales diferencias de los distintos formatos de sistemas de archivos.
- ☒ Conocer los sistemas de archivos virtuales.
- ☒ Crear y comprobar un sistema de archivos.
- ☒ Crear y administrar una zona de intercambio (swap).
- ☒ Administrar el montado automático de sistemas de archivos en el arranque.
- ☒ Configurar un automontaje.
- ☒ Saber cómo funciona el servicio udev.
- ☒ Configurar un disco duro con hdparm.
- ☒ Archivar datos.
- ☒ Copiar o sincronizar datos con rsync.
- ☒ Conocer los principales niveles de RAID.
- ☒ Crear y administrar discos en RAID por software.
- ☒ Crear y explotar volúmenes lógicos.
- ☒ Ampliar y reducir volúmenes lógicos.
- ☒ Realizar un volumen lógico de snapshot.
- ☒ Encriptar sistemas de archivos.
- ☒ Realizar copias de seguridad de sistemas de archivos.

Administración y configuración de sistemas de archivos

1. Administración de sistemas de archivos

a. Los sistemas de archivos más comunes

En la mayoría de los casos un sistema operativo se instala en un disco duro o en un periférico de almacenamiento similar. Si se examina detalladamente un disco duro nuevo se puede comprobar que su espacio de almacenamiento está formado por una serie de bytes sin ningún tipo de organización. Para usar adecuadamente la totalidad o parte de este espacio de almacenamiento, en primer lugar conviene segmentarlo, esto es el particionamiento, para después crear en las particiones operativas un sistema de archivos.

El sistema de archivos sirve para organizar un espacio de almacenamiento bruto, como una partición de disco para almacenar datos. Aunque normalmente se dice que se va a dar formato a un espacio de almacenamiento, en los entornos Linux se dice que se va a crear un sistema de archivos.

El término sistema de archivos viene del vocablo inglés filesystem. Usaremos el término "sistema de archivos" cuando se trate de un sistema de archivos añadido a un periférico de almacenamiento único o para designar el espacio de almacenamiento organizado, esté compuesto por uno o por muchos sistemas de archivos.

Existen muchos tipos de sistemas de archivos de los cuales los más comunes en entornos Linux son ext, reiserfs y xfs. Su estudio es necesario para obtener la certificación LPI.

ext

ext es el sistema de archivos histórico de los sistemas Linux. El formato original está en desuso y actualmente se considera a ext2 la versión base de ext. Permite un tamaño máximo de archivos de 2 TB y un tamaño de volumen máximo de 32 TB. ext2 ya no se despliega por defecto en los sistemas operativos modernos pero sigue estando disponible de forma universal. Gestiona fechas de archivos hasta el año 2038.

ext3 es una evolución de ext2 a la que se le añadió un registro de transacciones. Cada operación de escritura se guarda en el registro, lo que permite limitar las operaciones de validación a solo los elementos modificados desde la última validación realizada. Los tamaños máximos de archivos y volúmenes son los mismos que los de ext2. La gestión de fechas de archivos también es hasta el año 2038.

ext4, disponible desde octubre de 2008, aporta rendimientos superiores a sus predecesores. Los tamaños máximos se amplían hasta llegar a 16 TB para los archivos y 1 EB para los volúmenes. La asignación de espacio se realiza mediante extents, que son bloques de datos contiguos que permiten, asignando por adelantado espacio al disco, limitar espectacularmente la fragmentación de archivos. Para asegurarse no estar limitado para la gestión de fechas, la fecha límite de un archivo se sitúa en el año 2514.

reiserfs

reiserfs es un sistema de archivos con registro que para algunas operaciones tiene un rendimiento ligeramente mejor que ext3. reiserfs se creó como competencia de ext. Antiguo sistema de archivos por defecto de las distribuciones Suse, reiserfs ahora está en vías de desaparecer. Se le suele reprochar según las condiciones de uso una cierta fragilidad o una falta de rendimiento global.

xfs

xfs es el sistema de archivos histórico de los servidores unix IRIX. Fue puesto bajo licencia GPL en el año 2000. Su buen rendimiento así como su compatibilidad con espacios de almacenamiento muy grandes (8 exabytes de tamaño máximo contra los 16 y 32 terabytes para reiserfs y ext3) lo convierten en un sistema de archivos interesante.

b. Sistemas de archivos virtuales o pseudofilesystems

Un sistema de archivos normal tiene como objetivo habilitar el uso de un espacio de almacenamiento físico para un usuario o una aplicación. Sin embargo, en los sistemas Linux hay sistemas de archivos virtuales que sólo están presentes en memoria sin ocupar espacio en disco. Los sistemas de archivos virtuales que hay que dominar para la certificación LPI son proc y sys.

proc

El sistema de archivos virtual proc, generalmente montado en el directorio /proc, permite la visualización de los elementos del sistema relacionados con la gestión de procesos por parte del núcleo. Además, proc muestra algunos datos sobre el hardware del sistema.

Visualización de los datos del procesador

A continuación se observan los datos técnicos relacionados con el procesador en uso. Se puede observar por ejemplo la velocidad real del reloj en el momento de la ejecución del comando, lo que demuestra la buena gestión de la energía en un sistema poco usado.

```
usuario@alfa:~$ cat /proc/cpuinfo
processor       : 0
vendor_id       : GenuineIntel
cpu family      : 6
model           : 42
model name      : Intel(R) Core(TM) i5-2500 CPU @ 3.30GHz
stepping        : 7
cpu MHz         : 3294.845
cache size      : 6144 KB
fdiv_bug        : no
hlt_bug         : no
f00f_bug        : no
coma_bug        : no
fpu             : yes
fpu_exception   : yes
cpuid level     : 5
wp              : yes
flags           : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr
pge mca cmov pat pse36 clflush mmx fxsr sse sse2 syscall nx
lm constant_tsc up pni monitor ssse3 lahf_lm
bogomips        : 6429.86
clflush size    : 64
power management:
```

sys

El sistema de archivos virtual **sys**, generalmente montado en el directorio /sys, permite visualizar elementos de sistema relacionados con los periféricos.

Visualización de capacidades hotplug de un disco duro

Hay muchos pseudoarchivos de /proc y /sys que tienen su contenido limitado a un solo carácter. En este caso 0 indica que el disco sda no se puede conectar en caliente.

```
usuario@alfa:~$ cat /sys/block/sda/removable
0
```

c. Creación de sistemas de archivos

El administrador crea los sistemas de archivos en espacios de almacenamiento sin tratar, tradicionalmente particiones de disco. A partir de este momento se revisarán ocasionalmente por el administrador o a intervalos regulares por el sistema. La creación de sistemas de archivos se realiza tradicionalmente con el comando **mkfs**.

Sintaxis del comando mkfs

```
mkfs -t tipo dispositivo
```

mkfs: opciones y parámetros	
-t *tipo*	Especificación del tipo de sistema de archivos que se creará. Valores que debemos saber: ext2, ext3, ext4, reiserfs, xfs.

dispositivo	Archivo especial de bloque que designa el dispositivo donde se creará el sistema de archivos.

d. Revisión de los sistemas de archivos

La revisión de un sistema de archivos consiste principalmente en la comprobación de coherencia entre su tabla de inodos y los bloques de datos correspondientes. Es decir, para cada inodo se comprobará que los bloques de datos apuntados por este inodo están presentes, con el número y la cantidad indicados. En el caso de los sistemas de archivos con registro la opción -f forzará una comprobación completa de un sistema de archivos que aparentemente ya esté limpio, como por ejemplo un sistema de archivos que no ha sufrido ninguna operación de escritura desde la última comprobación con éxito.

La revisión del sistema de archivos se realiza con el comando **fsck**.

Sintaxis del comando fsck

```
fsck -t tipo dispositivo
```

fsck: opciones y parámetros	
-t *tipo*	Tipo de sistema de archivos que se comprobará.
dispositivo	Archivo especial de bloque que designa el dispositivo en el que se encuentra el sistema de archivos que se desea revisar.

Aunque el comando fsck permite comprobar los sistemas de archivos xfs, se recomienda utilizar los comandos específicos xfs_check y xfs_repair.

e. Comandos específicos para sistemas de archivos ext

Las sintaxis mostradas a continuación para los comandos **mkfs** y **fsck** son universales y deben funcionar. No obstante, hay que saber que estos comandos invocan en realidad a otros subprogramas (mkfs.ext2 por ejemplo para mkfs -t ext2), y que además hay otros comandos específicos que producirán el mismo resultado (**mke2fs** es otra alternativa a **mkfs** -t ext2). La mayor parte de las preguntas de la certificación LPI usan esta sintaxis común.

Al contrario que con el comando fsck, e2fsck funciona por defecto en modo interactivo. Para un funcionamiento en modo no interactivo tiene que recibir como parámetro la opción -p. De este modo revisa automáticamente el sistema de archivos sin necesitar la intervención del usuario.

f. Creación de un sistema de archivos ext

El comando mke2fs permite crear directamente sistemas de archivos ext. El formato utilizado por defecto es ext2, aunque la opción -j (journal) permite la creación de estructuras de un sistema de archivos ext3.

Creación de un sistema de archivos ext2

```
mke2fs dispositivo
```

Creación de un sistema de archivos ext3

```
mke2fs -j dispositivo
```

```
mke2fs -text3 dispositivo
```

Creación de un sistema de archivos ext4

```
mke2fs -text4 dispositivo
```

Donde *dispositivo* representa el archivo especial en modo de bloques que identifica el periférico en el que se desea crear el sistema de archivos.

g. Consulta y modificación de sistemas de archivos ext

El comando tune2fs permite visualizar los parámetros de un sistema de archivos ext y, si se desea, modificar algunos de ellos.

Consulta de parámetros de un sistema de archivos ext con tune2fs

```
tune2fs -l dispositivo
```

Donde *dispositivo* representa el archivo especial en modo de bloques que identifica el periférico en el que se encuentra el sistema de archivos que queremos consultar.

La diferencia entre el formato ext2 y ext3 es la presencia o la ausencia de un registro de transacciones. El comando tune2fs permite añadir este registro a un sistema de archivos ext2, y por consiguiente convertirlo en ext3.

Visualización de parámetros de un sistema de archivos ext3

Se puede observar que el valor has_journal figura en la sección Filesystem features, lo que indica que se trata de un sistema de archivos de tipo ext3. Un sistema de archivos de tipo ext4 se caracteriza por la mención de huge_file.

```
alfa:/dev# tune2fs -l /dev/hda1
tune2fs 1.41.3 (12-Oct-2008)
Filesystem volume name:   <none>
Last mounted on:          <not available>
Filesystem UUID:          4c98c60d-d719-418e-aeda-cefe56fd40b8
Filesystem magic number:  0xEF53
Filesystem revision #:    1 (dynamic)
Filesystem features:      has_journal ext_attr resize_inode
dir_index filetype needs_recovery sparse_super large_file
Filesystem flags:         signed_directory_hash
Default mount options:    (none)
Filesystem state:         clean
Errors behavior:          Continue
Filesystem OS type:       Linux
Inode count:              498736
Block count:              1994060
Reserved block count:     99703
Free blocks:              1795699
Free inodes:              475810
First block:              0
Block size:               4096
Fragment size:            4096
Reserved GDT blocks:      486
Blocks per group:         32768
Fragments per group:      32768
Inodes per group:         8176
Inode blocks per group:   511
Filesystem created:       Fri Jul  1 06:10:07 2011
Last mount time:          Fri Jul  1 18:06:07 2011
Last write time:          Fri Jul  1 18:06:07 2011
Mount count:              3
Maximum mount count:      25
Last checked:             Fri Jul  1 06:10:07 2011
Check interval:           15552000 (6 months)
Next check after:         Wed Dec 28 05:10:07 2011
Reserved blocks uid:      0 (user root)
Reserved blocks gid:      0 (group root)
First inode:              11
Inode size:               256
Required extra isize:     28
Desired extra isize:      28
Journal inode:            8
Default directory hash:   half_md4
Directory Hash Seed:      e4f96d72-4192-4a0c-b565-7cb2aa44522c
Journal backup:           inode blocks
```

```
alfa:/dev#
```

Las utilidades dumpe2fs, debugfs o debugreiserfs permiten obtener más información de bajo nivel sobre los sistemas de archivos. Su conocimiento en detalle no se requiere para la certificación LPI.

Conversión de un sistema de archivos ext2 a ext3 con tune2fs

```
tune2fs -j dispositivo
```

Conversión de un sistema de archivos ext3 a ext4 con tune2fs

```
tune2fs -0 extents,uninit_bs,div_index dispositivo
```

Donde *dispositivo* representa el archivo especial de bloque que identifica el periférico en el que se encuentra el sistema de archivos que queremos modificar.

Les options activées sont celles correspondant aux nouvelles fonctions apportées par ext4.

h. Identificación de sistemas de archivos

Algunos parámetros de los sistemas de archivos se pueden modificar después de su creación. Entre estos parámetros, algunos van tomando cada vez más importancia en los sistemas Linux modernos y simplificarán (posiblemente) las operaciones de montaje. Estos parámetrosson la**etiqueta** o **label** y el **uuid**. Ambos permiten montar sistemas de archivos locales sin tener que identificarlos por su archivo de bloque especial, /dev/sdb1 por ejemplo. Aunque esta evolución no se ve necesariamente como un progreso o una simplificación por parte de todos, su generalización así como su presencia en los exámenes LPI la convierten en un apartado de obligado estudio.

La etiqueta de un sistema de archivos

Tal y como su nombre indica, la etiqueta (label en inglés) es un nombre que se asigna al sistema de archivos para identificarlo de forma natural. La etiqueta tiene que ser facilitada por el administrador bien cuando se crea el sistema de archivos o bien después mediante un comando de tunning. Los sistemas inspirados en Red Hat son los principales usuarios de etiquetas.

Asignar una etiqueta a un sistema de archivos existente	
tune2fs -L *etiqueta dispositivo*	Asigna una etiqueta al dispositivo de almacenamiento*dispositivo*.
reiserfstune -l *etiqueta dispositivo*	Asigna una etiqueta al dispositivo de almacenamiento*dispositivo*.
xfs_admin -L *etiqueta dispositivo*	Asigna una etiqueta al dispositivo de almacenamiento*dispositivo*.

El UUID de un sistema de archivos

El UUID (*Universally Unique Identifier*), como sucede con la etiqueta, permite asociar a un periférico de almacenamiento un identificador en vez de usar el archivo de bloque especial (/dev/sdb1 por ejemplo). La diferencia con la etiqueta es que la asignación del uuid es automática cuando se crea el sistema de archivos. Sin embargo, se puede modificar posteriormente con los comandos de tunning de los sistemas de archivos. Cada vez más distribuciones generalizan el uso del uuid. Éste es el caso en particular de las distribuciones ubuntu.

Si no se sabe cómo determinar el UUID de un nuevo sistema no hay por qué preocuparse, generalmente se crea de forma aleatoria y su tamaño (128 bits) es garantía de su unicidad (probable).

Modificación de un uuid en un sistema de archivos previamente creado	

En la tabla

tune2fs -U *uuid* *dispositivo*	Asignación del UUID *uuid* al dispositivo de almacenamiento *dispositivo*.
tune2fs -U random*dispositivo*	Asignación de un UUID aleatorio al dispositivo de almacenamiento *dispositivo*.
tune2fs -U time*dispositivo*	Asignación de un UUID basado en la hora de creación al dispositivo de almacenamiento *dispositivo*.
reiserfstune -u *uuid* *dispositivo*	Asignación del UUID *uuid* al dispositivo de almacenamiento *dispositivo*.
xfs_admin -U *uuid* *dispositivo*	Asignación del UUID *uuid* al dispositivo de almacenamiento *dispositivo*.

anterior *dispositivo* representa el archivo especial de bloque que representa al dispositivo que alberga el sistema de archivos que se va a modificar, por ejemplo /dev/sda3.

2. Administración del swap

a. ¿Por qué usar el swap y en qué cantidad?

El swap o memoria de intercambio es un espacio de almacenamiento usado para subsanar la falta de memoria física en el sistema. Cuando la memoria física se vuelve un bien escaso para las aplicaciones, una parte de la información almacenada en memoria y que no se ha utilizado recientemente se mueve al espacio de swap, liberando espacio para las aplicaciones que necesitan memoria inmediatamente. Si una aplicación requiere el uso de algún dato que ha sido movido a la zona de swap, el mecanismo de swap es de nuevo el encargado de liberar todavía más espacio en memoria física para volver a traer los datos desde el intercambio para que se puedan utilizar.

No hay que confundirse sobre el uso que hay que hacer del swap. En un funcionamiento normal, un servidor o una estación de trabajo Linux no debería usar el swap. La gran época del swap se dio cuando la memoria tenía un coste tan caro que era necesario encontrar para el servidor un compromiso entre el coste del sistema y el rendimiento que podía obtener. Hoy en día el coste relativamente bajo de la memoria hace que un sistema no tenga que recurrir tan a menudo al mecanismo del swap. Sólo sucede en casos de un consumo en exceso accidental de memoria. El swap es por lo tanto un tipo de memoria de repuesto que evita el cuelgue de un servidor gestionando el balance entre las necesidades de memoria y los recursos disponibles.

La cantidad de memoria dedicada al swap depende a menudo del autor, la fuente y la época. Es difícil, cuando se está instalando de forma manual un sistema, tomar una decisión serena. Hay cierto consenso en torno a valores comprendidos entre una y dos veces el tamaño de la memoria RAM. De todos modos, las instalaciones por defecto de las distribuciones generalmente proponen la creación automática de un espacio de swap. Para una instalación a medida, los valores comunes (de una a dos veces la memoria RAM) son perfectamente aceptables y en caso de duda, como el espacio en disco es todavía más barato, lo mejor es sobredimensionar.

b. Optimización del swap

El swap se puede optimizar tanto en cantidad como en calidad. Puede suceder que durante la instalación se haya decidido un tamaño de swap demasiado pequeño: por ejemplo, cuando se instala en un servidor una aplicación que exige una cantidad de RAM y un swap diez veces superior a los existentes.

Además, el espacio de intercambio puede trasladarse a un disco más rápido: una SAN o una bahía de disco más moderna y por lo tanto más rápida que en el que está instalado el sistema, con lo que el uso del swap podría ser más rápido en estos sistemas de almacenamiento.

Por estas razones puede ser útil crear un nuevo espacio de swap que se añadirá o substituirá al espacio inicial.

Naturaleza del espacio de swap

Se puede crear el swap desde varios espacios de almacenamiento pudiendo ser particiones o archivos. Como nos interesa que el núcleo acceda directa y exclusivamente al swap, es preferible en términos de rendimiento usar particiones en lugar de archivos de intercambio, ya que el sistema de archivos representa un intermediario adicional para acceder al almacenamiento físico.

Si el swap se ubica en una partición, ésta tiene que crearse de tipo 82 con una herramienta de particionado adecuada (fdisk Linux, por ejemplo). Si está en un archivo, tiene que estar siempre disponible en un sistema de archivos montado a perpetuidad.

Creación del espacio de swap

Para poder usar el espacio de swap, hay que prepararlo. Para ello hay que hacer como cuando se crea un sistema de archivos en un espacio de almacenamiento sin tratar. Esta preparación se realiza mediante el comando **mkswap**, y puede aplicarse tanto a una partición como a un archivo de tamaño determinado.

Sintaxis del comando mkswap

```
mkswap espacio_de_almacenamiento
```

Donde *espacio_de_almacenamiento* representa la ubicación física del espacio de swap cuya denominación puede realizarse de distintos modos:

Posibles denominaciones de los espacios de almacenamiento para el comando mkswap	
/ruta/archivo	Estructura el archivo para que pueda usarse como espacio de swap.
/dev/device	Estructura el espacio de almacenamiento designado por el archivo especial de bloque para que se pueda usar como espacio de swap.
-L LABEL	Estructura el espacio de almacenamiento designado por la etiqueta LABEL para que se pueda usar como espacio de swap.
-U UUID	Estructura el espacio de almacenamiento designado por el uuid UUID para que se pueda usar como espacio de swap.

Uso del swap

Una vez que el espacio de swap ha sido creado, se tiene que dejar disponible al núcleo mediante el comando **swapon**. Entonces el sistema ya será capaz de realizar intercambios desde el nuevo espacio creado.

Sintaxis del comando swapon para activar el espacio de swap

```
swapon espacio_de_almacenamiento
```

Donde *espacio_de_almacenamiento* representa la ubicación física del espacio de intercambio cuya denominación puede realizarse de distintos modos:

Posibles denominaciones de los espacios de almacenamiento para el comando swapon	
/ruta/archivo	Habilita la utilización del archivo como swap para el núcleo.
/dev/device	Habilita la utilización del espacio de almacenamiento designado por el archivo especial de bloque como swap para el núcleo.
-L LABEL	Habilita la utilización del espacio de almacenamiento designado por la etiqueta LABEL como swap para el núcleo.
-U UUID	Habilita la utilización del espacio de almacenamiento designado por el uuid UUID como swap para el núcleo.

Desactivación de un espacio de swap

Si desea que el sistema deje de usar un espacio de swap, hay que indicárselo con el comando**swapoff**.

Sintaxis del comando swapoff para desactivar un espacio de swap

```
swapoff espacio_de_almacenamiento
```

Donde *espacio_de_almacenamiento* representa la ubicación física del espacio de swap cuya denominación

puede realizarse de distintos modos:

Posibles denominaciones de los espacios de almacenamiento para el comando swapoff	
/ruta/archivo	Detiene el uso del espacio de swap en el archivo.
/dev/dispositivo	Detiene el uso del espacio de swap en el dispositivo.
-L LABEL	Detiene el uso del espacio de swap cuya etiqueta es LABEL.
-U UUID	Detiene el uso del espacio de swap cuyo uuid es UUID.

Visualización de los espacios de swap

El conjunto de espacios de swap usados, así como su naturaleza (archivo o partición), se pueden visualizar mediante el comando **swapon** explicado anteriormente.

Sintaxis del comando swapon para visualizar la configuración del swap

```
swapon -s
```

Ejemplo de uso del comando swapon

El comando indica la partición o el archivo usado, el tamaño reservado y la cantidad de swap usado.

Otra visualización de swap

```
alfa:~# swapon -s
Filename    Type        Size      Used  Priority
/dev/sda5   partition   409616    0  -1
```

También se puede visualizar la configuración del swap consultando el contenido del archivo swaps del sistema de archivos virtual /proc.

```
alfa:~# cat /proc/swaps
Filename    Type        Size      Used  Priority
/dev/sda5   partition   409616    0  -1
```

3. Montaje de sistemas de archivos

a. Montaje y desmontaje

El comando **mount** permite montar el sistema de archivos de un periférico de almacenamiento en un directorio local. Generalmente, este directorio suele dejarse vacío. Como mínimo hay que pasarle por parámetro al comando **mount** el periférico que alberga el sistema de archivos y el directorio que constituirá su punto de montaje.

El comando **umount** realiza la operación inversa. Acepta como argumento el punto de montaje o el periférico físico que se desmontará.

Montaje de un sistema de archivos

```
mount -t tipo_fs -o opciones dispositivo punto_de_montaje
```

comando mount: opciones y parámetros	
tipo_fs	Opcional: tipo de sistema de archivos que se desea montar.
opciones	Opcional: opciones de montaje.
dispositivo	El periférico que alberga el sistema de archivos que queremos

Las opciones más comunes

	montar, identificado con un archivo especial de bloque.
punto_de_montaje	El directorio que servirá de punto de montaje para el sistema de archivos montado.

son **ro** (sólo lectura), **sync** (escrituras síncronas sin pasar por una caché en memoria) y **loop** (montaje de datos de archivos en lugar del sistema de archivos).

El montaje de un sistema de archivos con la opción sync permite substituir el uso de la caché por escrituras directas a disco y, de este modo, hacer más fiables las operaciones de escritura. El comando sync permite vaciar puntualmente la caché de un sistema de archivos que no se haya montado con esta opción de montaje.

Desmontaje de un sistema de archivos

```
umount -O opciones dispositivo punto_de_montaje
```

comando umount: opciones y parámetros	
opciones	Opcional: opciones de desmontaje.
dispositivo	Opcional si el punto de montaje se ha definido: el periférico que se desea desmontar.
punto_de_montaje	Opcional si el periférico se ha definido: el directorio que sirve de punto de montaje que queremos liberar.

Las opciones más comunes son **-f** (force: forzar el desmontaje) y **-l** (lazy: desmontaje perezoso que se hará efectivo cuando todos los recursos utilizados por el montaje hayan sido liberados).

El desmontaje de un sistema de archivos es necesario para realizar su revisión mediante el comando e2fsck. Por definición, el sistema de archivos montado en / no se puede desmontar debido a que está siempre ocupado. Se puede forzar la comprobación antes del montaje cuando arranca el sistema desde el comando **shutdown**.

Revisión del sistema de archivos raíz antes del montaje

```
shutdown -F -r now
```

b. Visualización de los sistemas de archivos montados

El comando **mount** sin argumentos permite visualizar los sistemas de archivos montados.

Además, cada montaje con éxito genera su línea correspondiente en el archivo **/etc/mtab**. La visualización del archivo **/proc/mounts** devuelve la misma información.

c. Archivo fstab

El archivo **/etc/fstab** permite definir los sistemas de archivos o los espacios de swap que se montarán automáticamente en el arranque. De forma opcional también permite definir sistemas de archivos que es posible que se monten, como por ejemplo los periféricos extraíbles. La sintaxis del comando **mount** invocado puntualmente se simplificará bastante.

El archivo **/etc/fstab** tiene que tener en cada línea el conjunto de elementos necesarios para el montaje de un sistema de archivos. Estos elementos son el punto de montaje, la definición del espacio de almacenamiento y las opciones de montaje. Para los espacios de swap la definición del punto de montaje no es aplicable.

El archivo **/etc/fstab** se compone de una línea por cada sistema de archivos que se montará. A su vez, cada línea se compone de seis campos obligatorios.

Formato típico de una línea de declaración de montaje en /etc/fstab

```
fs puntodemontaje tipo opciones dump fsck
```

Los campos se separan por espacios o tabulaciones.

Archivo /etc/fstab: formato de las líneas de definición de montaje		
Número de campo	**Campo**	**Definición**
1	*fs*	Sistema de archivos, definido por su archivo especial de bloque, su etiqueta o su uuid.
2	*puntodemontaje*	Punto de montaje.
3	*tipo*	Tipo de sistema de archivos. Obligatoriamente swap para swap, auto o tipo real de sistema de archivos en caso contrario.
4	*opciones*	Opciones de montaje. De hecho, son las opciones admitidas por el comando mount.
5	*dump*	Opcional. Si el comando dump se utiliza para la copia de seguridad del sistema, este campo tiene que valer 1 para asegurar la copia de seguridad. Si no, su valor por defecto es 0.
6	*fsck*	Opcional. En caso de revisión automática del sistema de archivos en el arranque, indica en qué orden se tiene que realizar la comprobación. Valor obligatorio de 1 para el sistema de archivos montado en /, 2 para el resto. 0 para que la revisión no se realice nunca.

Ejemplo de archivo /etc/fstab en Ubuntu

Observe que los discos se identifican con su uuid.

```
# /etc/fstab: static file system information.
#
# Use 'blkid -o value -s UUID' to print the universally unique identifier
# for a device; this may be used with UUID= as a more robust way to name
# devices that
```

Copias de seguridad

La gestión de las copias de seguridad es uno de los aspectos mejor tratados en los sistemas Linux. Desde las herramientas históricas que gestionaban en el mejor de los casos una cinta magnética hasta las herramientas modernas más sofisticadas y los programas de copia de seguridad comerciales, el abanico es muy amplio. Lo importante es conocer los medios disponibles y adaptar la estrategia de copia de seguridad a las necesidades en función del tiempo y el dinero que se esté dispuesto a invertir en las copias de seguridad.

1. Las herramientas de archivado

Las herramientas de archivado permiten realizar copias de seguridad más simples y, gracias a su simplicidad, sin duda más fiables. Su principio es sencillo: enviar un conjunto de archivos (en general una estructura de carpetas y subcarpetas) a un archivo, ya sea un archivo ordinario o un archivo especial que identifique a un periférico de almacenamiento.

a. El comando tar

El comando **tar**, de uso universal en los entornos Linux, se tiene que conocer con detalle. Su riqueza funcional puede impresionar, pero aunque el comando **tar** presenta muchas opciones, las que realmente se usan en la mayoría de los casos son menos de una decena.

Sintaxis del comando tar para crear un archivo

```
tar acción compresión nivel_de_información -f archivo_recopilación directorio
```

Sintaxis del comando tar para listar o extraer un archivo

```
tar acción compresión nivel_de_información -f archivo_recopilación
```

Comando tar: opciones y parámetros		
acción	-c	Crea un archivo de recopilación. Por lo tanto, será necesario indicar con un último parámetro el directorio a partir del cual se creará el archivo.
	-t	Lista el contenido de un archivo de recopilación existente.
	-x	Extrae el contenido de un archivo de recopilación existente en el directorio actual.
compresión		Sin compresión en el archivo de recopilación.
	-z	Compresión en formato gzip del archivo de recopilación.
	-j	Compresión en formato bz2 del archivo de recopilación.
nivel_de_información		Sin información, visualización mínima.
	-v	Con información, visualización detallada.
archivo_recopilación		El archivo que recibe o alberga el archivo. Este archivo puede ser un archivo especial de bloque o de carácter. Hoy en día casi siempre se usa con archivos normales.
directorio		En caso de creación de archivo de recopilación, determina el directorio a partir del cual se creará el archivo.

Aunque no sea obligado, es costumbre ponerle una extensión ".tar" a los archivos de recopilación tar, seguido de una extensión relacionada con el modo de compresión ".gz" o ".bzip2".

En caso de usar del comando **tar** para crear copias de seguridad, se redirigirá el archivo de copia de seguridad a un periférico extraíble o a un espacio de almacenamiento remoto.

Ejemplos de uso del comando tar

En estos ejemplos se crea un archivo tar comprimido a partir de un directorio, se borra el directorio y finalmente se restaura a partir del archivo.

```
A:~# ls
ejemplo
A:~# # nota: creación del archivo
A:~# tar czf copiaseguridad.tar.gz ejemplo
A:~# ls
copiaseguridad.tar.gz  ejemplo
A:~# # nota: destrucción del directorio
A:~# rm -r ejemplo
A:~# ls
copiaseguridad.tar.gz
A:~# # nota: restauración del archivo
A:~# tar xzf copiaseguridad.tar.gz
A:~# ls
copiaseguridad.tar.gz  ejemplo
A:~#
```

Si el comando **tar** se emplea para crear un archivo en cinta magnética y no sobre disco se recomienda no utilizar ningún tipo de compresión. El formato comprimido impediría una recuperación parcial de los datos en caso de deterioro de la cinta.

b. El comando cpio

El comando **cpio**, cuyo uso tiende a desaparecer en los entornos Linux, permite realizar archivos de recopilación no comprimidos a partir de un conjunto de archivos y directorios.

cpio tiene un uso particularmente poco intuitivo y generalmente no se usa, salvo en casos específicos. El problema de **cpio** es que este comando no permite que se le definan los elementos de los que se quiere generar una copia de seguridad por parámetro tal y como se hace con el comando **tar**. Hay que indicarle estos elementos en forma de lista de archivos por su entrada estándar. Del mismo modo, todas las operaciones en salida se realizan por redirección de la salida estándar. El comando **cpio** ha sobrevivido a pesar de estas desventajas de otro tiempo, y lo ha hecho gracias a estas limitaciones sintácticas: la lista de archivos de los que se desea guardar una copia de seguridad se proporciona casi siempre mediante una redirección del resultado del comando **find**. Sin embargo, sabemos desde el nivel 1 de la certificación LPI que el comando find es capaz de realizar búsquedas extremadamente precisas a partir de muchos criterios. Por lo tanto, es en aquellos casos donde queramos realizar copias de seguridad muy selectivas en los que se utilizará **cpio**.

Sintaxis del comando cpio para crear un archivo

```
find directorio criterio -print | cpio opciones > archivo_recopilación
```

Sintaxis del comando cpio para listar o extraer un archivo

```
cpio opciones < archivo_recopilación
```

Comando cpio: opciones y parámetros		
directorio		El directorio base a partir del que se realiza la búsqueda.
criterio		Criterios de búsqueda según la sintaxis del comando find.
opciones	-o	Modo copy-out. Indica que se está en modo de creación del archivo de recopilación. Si se usa, no se pueden usar las opciones i y t.
	-t	Asociada a la opción i, lista el contenido de un archivo de recopilación existente. No se puede usar junto con la opción o.
	-i	Modo copy-in. Indica que se está en modo de extracción o de consulta de un archivo. No se puede usar junto con la opción

Ejemplo de uso del comando cpio

Aunque es

		o.
	-v	Opcional: hace que el comando sea más locuaz.
archivo_recopilación		El archivo (especial o normal) que recibirá el archivo de recopilación.

imprescindible saber usar el comando tar de forma natural, no sucede lo mismo con cpio, no se puede recordar de forma razonable la sintaxis de este comando.

```
A:~# ls
ejemplo
A:~# # nota: creación del archivo
A:~# find ejemplo -print | cpio -o > archivo.cpio
1 block
A:~# ls
archivo.cpio  ejemplo
A:~# # nota: destrucción del directorio ejemplo
A:~# rm -rf ejemplo
A:~# # nota: restauración del archivo
A:~# cpio -i < archivo.cpio
1 block
A:~# ls
archivo.cpio  ejemplo
A:~#
```

2. Copias de seguridad a nivel de sistema de archivos

a. Copias de seguridad de sistemas de archivos ext

Las utilidades **dump** y **restore** permiten realizar copias de seguridad incrementales y la restauración de un sistema de archivos entero.

Ejemplo de copia de seguridad con dump

En este

```
root@alfa:/mnt# dump 0 -f /root/cs /dev/sdb1
  DUMP: Date of this level 0 dump: Sun Sep 30 17:01:24 2012
  DUMP: Dumping /dev/sdb1 (an unlisted file system) to cs
  DUMP: Label: none
  DUMP: Writing 10 Kilobyte records
  DUMP: mapping (Pass I) [regular files]
  DUMP: mapping (Pass II) [directories]
  DUMP: estimated 56 blocks
  DUMP: Volume 1 started with block 1 at: Sun Sep 30 17:01:25 2012
  DUMP: dumping (Pass III) [directories]
  DUMP: dumping (Pass IV) [regular files]
  DUMP: Closing /root/cs
  DUMP: Volume 1 completed at: Sun Sep 30 17:01:25 2012
  DUMP: Volume 1 50 blocks (0.05MB)
  DUMP: finished in less than a second
  DUMP: Date of this level 0 dump: Sun Sep 30 17:01:24 2012
  DUMP: Date this dump completed:  Sun Sep 30 17:01:25 2012
  DUMP: Average transfer rate: 50 kB/s
  DUMP: DUMP IS DONE
root@alfa:/mnt#
```

ejemplo, 0 es el nivel de la copia de seguridad (0 para copia de seguridad completa, n para cada número de incremento); la opción -f indica el archivo de destino de la copia de seguridad y finalmente el último parámetro es el archivo de bloque especial del sistema de archivos que se desea guardar.

Ejemplo de restauración con restore

```
root@alfa:~# mount /dev/sdb1 /mnt/data
root@alfa:~# cd /mnt/data
root@alfa:/mnt/data# restore -rvf /root/svg
Verify tape and initialize maps
Input is from a local file/pipe
Input block size is 32
Dump date: Sun Sep 30 17:01:24 2012
Dumped from: the epoch
Level 0 dump of an unlisted file system on alfa:/dev/sdb1
Label: none
Begin level 0 restore
Initialize symbol table.
Extract directories from tape
Calculate extraction list.
Make node ./lost+found
Extract new leaves.
Check pointing the restore
extract file ./a
extract file ./b
extract file ./c
Add links
Set directory mode, owner, and times.
Check the symbol table.
Check pointing the restore.
root@alfa:/mnt/data#
```

La opción r utilizada aquí indica que se realiza una restauración (y no otra operación como una comparación, por ejemplo); v permite que el comando sea un poco más locuaz y f anuncia el archivo de copia de seguridad a restaurar. La restauración de archivos se realiza en el directorio actual.

b. Copias de seguridad de sistemas de archivos xfs

Las utilidades **xfsdump** y **xfsrestore** permiten realizar copias de seguridad incrementales y restauraciones de un sistema de archivos xfs entero. Las características del sistema de archivos xfs que se desea guardar pueden mostrarse por el comando **xfs_info** según se necesite.

Ejemplo de copia de seguridad con xfsdump

```
root@alfa:/mnt# xfsdump -f cs /dev/sdb2
xfsdump: using file dump (drive_simple) strategy
xfsdump: version 3.0.4 (dump format 3.0) - Running single-threaded

 ============================ dump label dialog ==============================

please enter label for this dump session (timeout in 300 sec) -> label-session
session label entered: "label-session"

 ---------------------------- end dialog -----------------------------

xfsdump: WARNING: most recent level 0 dump was interrupted, but not resuming
that dump since resume (-R) option not specified
xfsdump: level 0 dump of alfa:/mnt/web
xfsdump: dump date: Sun Sep 30 17:05:54 2012
xfsdump: session id: 023ee21-b2a2-5deb-a901-aa43889ef9ee2
xfsdump: session label: "label-session"
xfsdump: ino map phrase 1: constructing initial dump list
xfsdump: ino map phrase 2: skipping (no pruning necessary)
xfsdump: ino map phrase 3: skipping (only one dump stream)
xfsdump: ino map construction complete
xfsdump: estimated dump size: 35401 bytes

 ============================ media level dialog ==============================

please enter label for media in drive 0 (timeout in 300 sec) -> label-media
media label entered: "label-media"
```

La opción -f anuncia el archivo de destino de la copia de

```
---------------------------- end dialog -----------------------------

xfsdump: creating dump session media file 0 (media 0, file 0)
xfsdump: dumping ino map
xfsdump: dumping directories
xfsdump: dumping non-directory files
xfsdump: ending media file
xfsdump: media file size 32654 bytes
xfsdump: dump size (non-dir files): 8926 bytes
xfsdump: dump complete: 10 seconds elapsed
xfsdump: Dump Status: SUCCESS
root@alfa:/mnt#
```

seguridad, cs es el archivo objetivo, y sdb2 el sistema de archivos que hay que guardar.

Ejemplo de restauración con xfsrestore

```
root@alfa:/mnt# xfsrestore -f cs web
xfsrestore: using file dump (drive_simple) strategy
xfsrestore: version 3.0.4 (dump format 3.0) - Running single-threaded
xfsrestore: searching media for dump
xfsrestore: examining media file 0
xfsrestore: dump description:
xfsrestore: hostname: alfa
xfsrestore: mount point: /mnt/web
xfsrestore: volume: /dev/sdb2
xfsrestore: session time: Sun Sep 30 17:05:54 2012
xfsrestore: level: 0
xfsrestore: session label: "label-session"
xfsrestore: media label: "label-media"
xfsrestore: file system id:
xfsrestore: session id: 023ee21-b2a2-5deb-a901-aa43889ef9ee2
xfsrestore: media id:
xfsrestore: using online session inventory
xfsrestore: searching media for directory dump
xfsrestore: reading directories
xfsrestore: 1 directories and 4 entries processed
xfsrestore: directory post-processing
xfsrestore: restoring non-directory files
xfsrestore: restore complete: 0 seconds elapsed
xfsrestore: Restore Status: SUCCESS
root@alfa:/mnt#
```

La opción -f anuncia el archivo que contiene la copia de seguridad y web es el directorio de destino de la restauración.

3. Los programas de copias de seguridad

a. AMANDA

AMANDA (*Advanced Maryland Automatic Network Disk Archiver*) es una solución de copias de seguridad creada inicialmente por la universidad de Maryland con licencia BSD. Disponible con licencia libre (gratuita) o comercial, AMANDA permite hacer copias de seguridad de forma local o en red, en discos o en cintas, de los datos de sistemas Linux/Unix o Windows.

b. Bacula

Bacula es un programa con licencia GPL que permite crear copias de seguridad de forma local o en red, en discos o en cintas, de los datos de sistemas Linux/Unix o Windows.

c. BackupPC

BackupPC es un programa con licencia GPL que permite crear copias de seguridad de forma local o en red, en discos o en cintas, de los datos de sistemas Linux/Unix o Windows.

d. Los programas comerciales

La mayoría de los grandes fabricantes de software de copia de seguridad soportan, a menudo de forma opcional, la creación de copias de seguridad de sistemas Linux. Por ello, habrá que instalar en cada sistema un agente de copia de seguridad que permitirá enviar los datos al servidor de copias de seguridad.

4. Duplicación y sincronización de datos

a. Copia binaria con dd

El comando de copia bloque a bloque dd permite realizar copias de bajo nivel de un periférico. Se utiliza especialmente para la duplicación de discos duros, pero también para la creación de imágenes binarias de periféricos de almacenamiento.

Sintaxis genérica del comando dd

```
dd if=entrada of=salida bs=tamaño_del_bloque count=número_de_bloques
```

Comando dd: opciones y parámetros	
entrada	El archivo que se copiará. Generalmente un archivo especial de bloque.
salida	El archivo destino de la copia, puede ser un archivo especial de bloque o un archivo normal.
tamaño_del_bloque	Opcional. Determina el tamaño de los bloques que se copiarán.
número_de_bloques	Opcional. El número de bloques que se desea copiar. Si se omite el parámetro la copia se detiene cuando ya no quedan más bloques para copiar.

Utilización del comando dd para una copia de disco duro

Copia del disco sdb al disco sdc.

```
root@servidor# dd if=/dev/sdb of=/dev/sdc
root@servidor#
```

Utilización del comando dd para realizar una imagen iso de un cdrom

El archivo iso generado se puede grabar con cualquier programa de grabación o se puede usar en una máquina virtual.

```
root@servidor# dd if=/dev/cdrom of=/home/usuario/imagen.iso
root@servidor#
```

Utilización del comando dd para borrar físicamente una memoria usb

Borrado físico de todos los bloques de una memoria usb visto como el dispositivo sdd. Atención, los datos no se podrán recuperar por ningún medio sencillo. ¡No debe equivocarse de disco!

```
root@servidor# dd if=/dev/zero of=/mnt/sdd
root@servidor#
```

Utilización del comando dd para crear un archivo vacío de 100 MB

Comando dd utilizado para crear un espacio de swap o generar grandes archivos para pruebas de copia.

```
root@servidor# dd if=/dev/zero of=/home/usuario/archivovacío bs=1024 count=100000
root@servidor#
```

b. Generación de un archivo iso con mkisofs

Los archivos iso son imágenes binarias de cdrom o dvdrom. Las imágenes iso se pueden montar con el comando **mount**, se pueden grabar con cualquier software de grabación y se pueden usar en máquinas virtuales que las interpretan como unidades de cdrom. Puede ser útil generar imágenes iso a partir de una estructura de directorios. Para ello, podemos usar el comando**mkisofs**.

Sintaxis del comando mkisofs

```
mkisofs -J -o imagen directorio
```

Comando mkisofs: opciones y parámetros	
-J	Opcional: genera registros Joliet además de la estructura de nombres iso9660. Mejora la compatibilidad con los sistemas Windows.
-o*imagen*	El archivo iso que se generará. Generalmente con la extensión ".iso".
directorio	El directorio a partir del cual se generará la imagen iso.

Ejemplo de uso del comando mkisofs

El archivo iso generado puede grabarse directamente con cualquier software de grabación.

```
pedro@ubuntu:~/Temp$ ls
data
pedro@ubuntu:~/Temp$ mkisofs -o imgcd.iso data
I: -input-charset not specified, using utf-8 (detected in locale settings)
Total translation table size: 0
Total rockridge attributes bytes: 0
Total directory bytes: 8192
Path table size(bytes): 50
Max brk space used 23000
178 extents written (0 MB)
pedro@ubuntu:~/Temp$ ls
imgcd.iso  data
pedro@ubuntu:~/Temp$ file imgcd.iso
imgcd.iso: ISO 9660 CD-ROM filesystem data 'CDROM                              '
pedro@ubuntu:~/Temp$
```

mkisofs es el nombre histórico del comando que permite crear archivos iso. Sin embargo, este comando se ha renombrado recientemente a genisoimage, y mkisofs está presente en las distribuciones recientes en forma de enlace simbólico a genisoimage.

La imagen iso generada es un archivo único en principio inaccesible si no se usa un software apropiado. De hecho, se puede montar el archivo de imagen como si de un periférico cualquiera se tratara.

Montaje local de una imagen iso

```
mount -o loop archivo_de_imagen punto_de_montaje
```

Donde *archivo_de_imagen* representa la imagen iso que se montará y *punto_de_montaje* el directorio que la albergará. La opción loop es imprescindible para el montaje de un archivo de imagen.

c. Sincronización de datos con rsync

En el marco de las estrategias de conservación de datos, puede ser útil replicar los datos de un servidor a otro, ya sea para garantizar la disponibilidad geográfica de datos idénticos o para prevenir un posible fallo de

un disco duro o de un servidor. El comando **rsync** cumple esta tarea a la perfección.

rsync ofrece muchos modos de funcionamiento, pero el más común en el marco de la sincronización de datos es el de disponer de un servicio **rsync** en un servidor y planificar sincronizaciones regulares desde las máquinas que contienen los datos que hay que replicar.

Para necesidades puntuales, el comando **rsync** puede usarse localmente para sincronizar dos estructuras de directorios. Su sintaxis es entonces equivalente a la del comando **cp**.

Configuración de un servidor rsync

La configuración se realiza mediante dos archivos: el archivo **/etc/default/rsync** que se deberá modificar y el archivo **/etc/rsyncd.conf** que se tendrá que crear.

Modificación del archivo /etc/default/rsync

```
RSYNC_ENABLE=true
```

Este parámetro permite el arranque automático o manual de **rsync** como servicio.

Creación del archivo /etc/rsyncd.conf

```
uid = usuario
read only = false
[instancia]
  path = directorio
```

Archivo /etc/rsyncd.conf: directivas y parámetros	
usuario	La cuenta de usuario que se utilizará para realizar las escrituras en el servidor.
read only = false	Imprescindible para que el servicio pueda escribir en el disco.
instancia	Nombre a su elección, habrá tantas instancias como clientes a replicar. Este nombre se usará en el cliente en el instante de la solicitud de sincronización.
directorio	Directorio donde se escribirán los datos sincronizados. La cuenta de usuario usada tiene que tener permisos de escritura en este directorio.

Después de configurarlo, hay que reiniciar el servicio **rsync** mediante el sistema habitual.

```
/etc/init.d/rsync restart
```

Sincronización de datos desde el cliente

La sincronización se realizará bajo demanda o mediante una tarea planificada con el comando**rsync**.

Sintaxis del comando rsync para una sincronización puntual

```
rsync -av --delete /directorio/ ip_servidor::instancia
```

Comando rsync: opciones y parámetros	
-a	Modo archivo: replica los datos de modo idéntico, conservando especialmente los permisos y los propietarios.
-v	Opcional: muestra detalladamente cada operación. Permite la visualización del progreso de la sincronización.
--delete	Copia replicada: los datos borrados en el cliente se borran también en el servidor.

directorio	El directorio de los datos locales que se replicarán.
instancia	El nombre de la instancia configurada en /etc/rsyncd.conf en el servidor.

Sincronización segura de datos con rsync

Si la sincronización de datos tiene que hacerse en un entorno hostil, se puede recurrir a SSH para el transporte de datos. En este modo de funcionamiento el daemon rsync no se ejecuta en el servidor y el ejecutable se inicia sobre la marcha por SSH para cada conexión entrante.

Sincronización segura con rsync

```
rsync -av --delete -e ssh directorio
usuario@dirección_servidor:/ruta_destino
```

rsync con ssh: opciones y parámetros	
-a	Modo archivo: replica los datos de modo idéntico, conservando especialmente los permisos y los propietarios.
-v	Opcional: muestra detalladamente cada operación. Permite la visualización del progreso de la sincronización.
--delete	Copia replicada: los datos borrados en el cliente se borran también en el servidor.
directorio	El directorio de los datos locales que se replicarán.
usuario	La cuenta de usuario existente en la máquina destino que se utilizará para la sesión ssh.
dirección_servidor	Dirección IP del servidor de destino.
ruta_destino	Directorio de destino para la sincronización de datos de la máquina objetivo.

Ejemplo de

sincronización segura

*El comando **rsync** permite crear una réplica entre discos de distintos sistemas con bajo coste.*

```
[root@beta data]# rsync -av --delete -e ssh /root/data
root@192.168.200.106:/root/svg
root@192.168.200.106's password:
building file list ... done
created directory /root/svg
data/
data/dos/
data/dos/archivo2
data/tres/
data/uno/
data/uno/archivo1

sent 50047 bytes  received 88 bytes  14324.29 bytes/sec
total size is 49785  speedup is 0.99
[root@beta data]#
[root@beta data]# rm -rf un
[root@beta data]# rsync -av --delete -e ssh /root/data
root@192.168.200.106:/root/svg
root@192.168.200.106's password:
building file list ... done
deleting data/uno/archivo1
deleting data/uno/
data/

sent 109 bytes  received 26 bytes  38.57 bytes/sec
total size is 35964  speedup is 266.40
[root@beta data]#
```

RAID

RAID (*Redundant Array of Independent Disks*, Conjunto redundante de discos independientes) es una tecnología de uso de discos duros que permite utilizar un espacio de almacenamiento repartido entre varios discos físicos con el objetivo de aumentar el rendimiento, la tolerancia a fallos o ambos. Aunque esta tecnología normalmente se gestiona por hardware en las bahías de los discos o en las SAN, también se puede delegar esta tarea a Linux. En este caso, el núcleo de Linux tendrá a su disposición varios discos duros y organizará los bloques de datos en estos discos para hacer las particiones lógicas que albergarán los sistemas de archivos.

Sólo tratamos los RAID gestionados por software por el núcleo de Linux. Para un servidor en producción, es probable que el RAID esté gestionado por una controladora hardware. En este caso, la controladora presentará unidades lógicas (LUN) que se verán como particiones normales y para el sistema es indiferente si la controladora trabaja en RAID o no.

1. Los principales niveles de RAID

a. RAID 0

RAID 0 tiene como único objetivo la rapidez de acceso a los datos y no gestiona de modo alguno la tolerancia a los fallos. Es muy importante saber que en RAID 0 el fallo de cualquiera de los elementos provoca la pérdida total de los volúmenes utilizados. El principio de RAID 0 es repartir los datos escritos en bloques y escribir los bloques al mismo tiempo en los discos físicos que componen el volumen RAID.

El espacio utilizable en un volumen RAID 0 es igual a la suma de los espacios de disco utilizados.

b. RAID 1

RAID 1, al contrario que RAID 0, no busca en ningún caso mejorar el rendimiento, sino únicamente asegurar los datos. En RAID 1, cada bloque de datos se duplica y se escriben tantas copias como discos hay en el volumen RAID. De este modo, si un disco falla, los datos siguen estando disponibles.

El espacio utilizable en un volumen RAID 1 es igual al espacio disponible en un solo disco.

c. RAID 5

RAID 5 reúne las ventajas de RAID 0 y de RAID 1. Hay que disponer de al menos 3 discos para configurarlo. Cuando se produce una operación de escritura en un volumen RAID 5, los bloques de datos se escriben usando todos los discos del volumen, a excepción de un bloque de paridad en un disco que se deduce a partir de los bloques de datos con un XOR ("o exclusivo"). En caso de fallo de un disco, los bloques de datos que faltan se recalculan realizando una operación XOR de todos los bloques restantes, datos y paridad.

El espacio explotable en un volumen RAID 5 es igual a la suma de los espacios de disco usados menos uno y menos un posible disco de reserva (spare).

2. Configuración de RAID

a. Creación de un volumen RAID

Los volúmenes RAID se configuran bastante fácilmente mediante el comando **mdadm**. Hay que disponer de varios espacios de almacenamiento, discos duros enteros o particiones, determinar el nivel de RAID deseado y elegir el nombre o el número del volumen que se desea crear.

El comando **mdadm** tiene su configuración, especialmente el orden de escaneo de todas las particiones encontradas en /proc/partitions, en su archivo de configuración /etc/mdadm/mdadm.conf. Generalmente no es necesario modificar la configuración por defecto.

Sintaxis del comando mdadm para la creación o la desactivación de un volumen RAID

```
mdadm acción volumen -l nivel -n número_de_discos almacenes
```

Creación de un volumen con mdadm: opciones y parámetros	
acción	-C: crea un volumen RAID.
	-S: desactiva un volumen y libera los recursos.
volumen	El archivo de bloque que se creará para representar el nuevo volumen. A menudo es /dev/mdx, pero puede tener un nombre cualquiera.
nivel	Valor del nivel de RAID, generalmente 0, 1 o 5.
número_de_discos	Número de espacios de almacenamiento que se van a usar, seguido de los archivos de bloque que representarán a estos espacios.
almacenes	Periféricos de almacenamiento separados por espacios e identificados por su archivo especial de bloque.

Ejemplo de la creación de un volumen RAID 1 en Debian

Se usan los dos discos duros /dev/sdb y /dev/sdc para crear el volumen RAID1

```
root@servidor# mdadm -C /dev/md0 -l 1 -n 2 /dev/sdb /dev/sdc
mdadm: array /dev/md0 started
root@servidor#
```

b. Comprobación de un volumen RAID

Es también el comando **mdadm** el que nos permitirá conocer el estado y la naturaleza de un volumen RAID desconocido.

Comprobación de un volumen RAID

```
mdadm -D volumen
```

Donde *volumen* es el archivo especial de dispositivo de bloque que representa el volumen RAID.

Ejemplo de comprobación de un volumen RAID

Es importante conocer y utilizar los comandos de diagnóstico para una buena administración y documentación del almacenamiento.

El

```
# mdadm -D /dev/md0
A:~# mdadm -D /dev/md0
/dev/md0:
        Version : 00.90
  Creation Time : Wed Jan 13 22:52:26 2010
     Raid Level : raid5
     Array Size : 4194176 (4.00 GiB 4.29 GB)
  Used Dev Size : 2097088 (2048.28 MiB 2147.42 MB)
   Raid Devices : 3
  Total Devices : 3
Preferred Minor : 0
    Persistence : Superblock is persistent

    Update Time : Wed Jan 13 22:54:49 2010
          State : clean, degraded, recovering
 Active Devices : 2
Working Devices : 3
 Failed Devices : 0
  Spare Devices : 1

         Layout : left-symmetric
     Chunk Size : 64K
```

```
 Rebuild Status : 90% complete

           UUID : a20a3883:3badc821:e24ccd6d:eee2883d (local to host A)
         Events : 0.4

    Number   Major   Minor   RaidDevice State
       0       8        0        0      active sync   /dev/sda
       1       8       16        1      active sync   /dev/sdb
       3       8       32        2      spare rebuilding   /dev/sdc
```

archivo **/proc/mdstat** también proporciona información sobre el estado de los discos RAID en un sistema Linux.

Ejemplo de archivo /proc/mdstat

El archivo mdstat proporciona una visualización resumida de los volúmenes RAID y de los discos que lo componen.

```
Personalities : [raid0]
md0 : active raid0 sdb[1] sda[0]
      4194176 blocks 64k chunks

unused devices: <none>
```

c. Uso de los volúmenes RAID

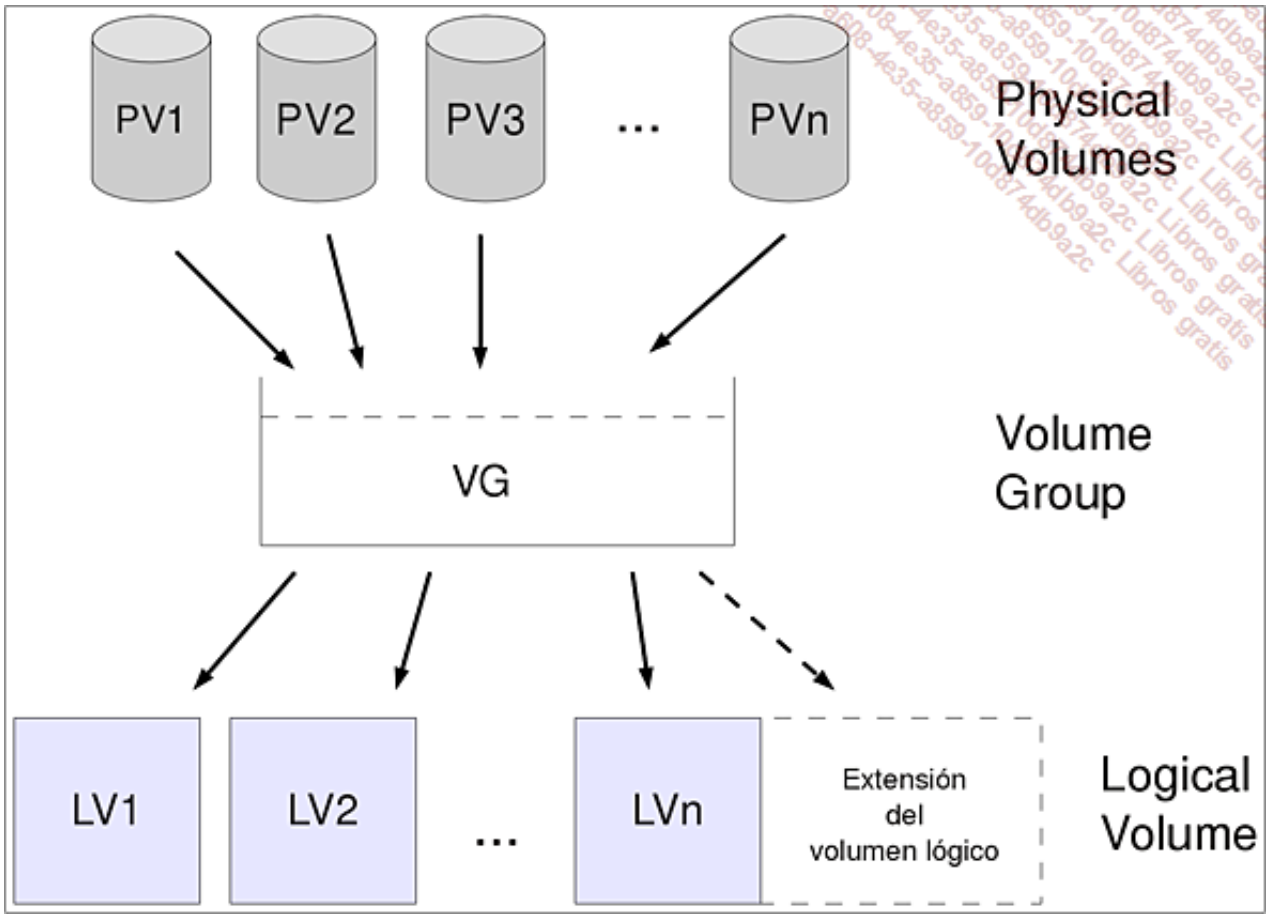
Una vez que se han creado los volúmenes con el comando **mdadm**, éstos se identifican por sus respectivos archivos de bloque especiales y soportarán la creación de un sistema de archivos así como el montaje, ya sea manual o invocado desde el archivo **/etc/fstab**.

Logical Volume Manager

El sistema de particionamiento tradicional de los discos impone ciertas restricciones tales como un número máximo de 4 particiones o el carácter obligatoriamente contiguo del espacio particionado. Aunque existe un gran número de utilidades que permiten redimensionar particiones sobre la marcha, no se puede extender una partición con espacio no contiguo, por ejemplo con otra unidad de disco duro. Para superar estas limitaciones la mayoría de los creadores de sistemas operativos han creado sistemas de administración de espacios de disco más o menos propietarios, como los discos dinámicos para Windows o los volúmenes NSS de Novell. Para los sistemas Linux la solución se llama Logical Volume Manager (administrador de volúmenes lógicos). Los volúmenes lógicos permiten crear un número ilimitado de volúmenes y extenderlos a voluntad, incluso a partir de espacios de almacenamiento que se encuentran en discos y controladoras distintos.

Es costumbre conservar los términos ingleses cuando se habla de elementos LVM, lo que ayudará especialmente a recordar de manera sencilla los comandos de uso. Algunos términos, como los Logical Volumes que admiten una traducción fácil y natural, se seguirán llamando en su forma inglesa.

1. Arquitectura de los volúmenes lógicos



Una arquitectura LVM se compone de **PV** (*Physical Volumes*), de **VG** (*Volume Groups*) y de **LV** (*Logical Volumes*).

Un volumen lógico es el equivalente funcional de una partición tradicional, se identifica por un archivo especial de bloque y albergará generalmente un sistema de archivos para ser montado.

Los Logical Volumes se componen de bloques de datos, extraídos en una capa de abstracción llamada Volume Group, que a su vez se alimenta de los espacios de almacenamiento brutos (discos o particiones) llamados Physical Volumes.

En una arquitectura LVM basada en varios volúmenes físicos el fallo en cualquiera de ellos provocará que todos los volúmenes lógicos que dependen de él queden fuera de servicio. Por tanto, es conveniente crear sólo volúmenes físicos para volúmenes con tolerancia a fallos como los que están en RAID, ya sea software o hardware.

2. Comandos LVM

Los comandos de administración del LVM se construyen con un prefijo que determina el objeto que se desea administrar junto a un sufijo que es la acción administrativa que se quiere realizar.

Construcción de los comandos LVM		
prefijo	**sufijo**	
pv	create	Creación de un elemento LVM.
vg	extend	Extensión de un VG o de un LV.
lv	reduce	Reducción de un VG o de un LV.
	display	Visualización de los datos de un elemento LVM.

a. Creación de elementos

Se comenzará creando los PV (*physical volumes*) a partir de espacios de almacenamiento. Pueden ser discos enteros o particiones tradicionales, cuyo tipo deberá modificarse a 8e. Cabe decir que la construcción de un PV a partir de particiones tradicionales se reserva generalmente a pruebas y que un uso en producción para volúmenes de datos se basa casi siempre en discos enteros.

Creación de volúmenes físicos

Los volúmenes físicos se crean con el comando **pvcreate**.

Sintaxis del comando pvcreate

```
pvcreate dispositivo
```

Donde *dispositivo* representa el archivo especial de bloque que alberga el volumen físico, ya sea disco o partición.

Creación de grupos de volúmenes

Los grupos de volúmenes se crean con el comando **vgcreate**.

Sintaxis del comando vgcreate

```
vgcreate nombre_vg dispositivo_pv
```

vgcreate: opciones y parámetros	
nombre_vg	Nombre del grupo de volúmenes. Valor a su elección.
dispositivo_pv	Archivo especial de bloque que alberga el o los pv que alimentan al vg.

El grupo de volúmenes creado de este modo aparecerá como un directorio, con el nombre del grupo de volúmenes creado, directamente en /dev. Atención, este directorio sólo aparecerá realmente cuando se haya creado un primer volumen lógico a partir del grupo de volúmenes.

Creación de volúmenes lógicos

Los volúmenes lógicos se crean con el comando **lvcreate**. Se pueden crear tantos volúmenes lógicos como se deseen siempre y cuando quede espacio disponible en el Volume Group.

Sintaxis del comando lvcreate

```
lvcreate -L tamaño -n nombre_lv nombre_vg
```

lvcreate: opciones y parámetros	
tamaño	Tamaño del volumen lógico. Es un valor numérico seguido directamente de la unidad.

El volumen lógico creado

nombre_lv	Nombre del volumen lógico. Valor a su elección.
nombre_vg	Nombre del grupo de volúmenes a partir del cual se creará el volumen lógico.

de este modo aparecerá como un archivo especial de bloque en el directorio que tiene el nombre de su grupo de volúmenes en /dev. Éste será el archivo especial que se usará para las operaciones de montaje.

b. Diagnósticos del LVM

Las arquitecturas LVM son a menudo desconcertantes, dado que hay que realizar un gran número de operaciones para realizar la creación de un volumen lógico. Además, aunque uno se figure bastante bien lo que puede ser un volumen físico, la naturaleza abstracta del grupo de volúmenes dificulta su comprensión. Por estas razones, es esencial hacerse una idea precisa del conjunto de elementos utilizados en una arquitectura LVM y documentarlos a conciencia. Por suerte, las herramientas de diagnóstico del LVM son precisas y permiten en cada fase comprobar que las operaciones se hayan desarrollado correctamente.

Visualización de la información sobre los volúmenes físicos

La información detallada de todos los volúmenes físicos presentes en un sistema se muestra mediante el comando **pvdisplay**. Si prefiere obtener información más resumida, puede probar con**pvs**.

Ejemplo de uso del comando pvdisplay

Es importante identificar los volúmenes físicos con el comando pvdisplay. La utilidad fdisk indicaría un disco sin tabla de particiones y daría a entender que es un disco virgen.

Ejemplo de uso del comando pvs

```
A:~# pvdisplay
  "/dev/sdb" is a new physical volume of "2,00 GB"
  --- NEW Physical volume ---
  PV Name               /dev/sdb
  VG Name
  PV Size               2,00 GB
  Allocatable           NO
  PE Size (KByte)       0
  Total PE              0
  Free PE               0
  Allocated PE          0
  PV UUID               UHSnwO-EKMh-QbDn-1qj0-f7Az-KKkx-3XcyZz
A:~#
```

Lo importante en dos líneas.

```
A:~# pvs
  PV         VG   Fmt  Attr PSize PFree
  /dev/sdb        lvm2 --   2,00G 2,00G
A:~#
```

Visualización de la información de los grupos de volúmenes

La información detallada de todos los grupos de volúmenes presentes en un sistema se muestra mediante el comando **vgdisplay**. Si desea obtener la información resumida, puede probar con **vgs**.

Ejemplo de uso del comando vgdisplay

La visualización de los detalles de los grupos de volúmenes permite conocer el tamaño total disponible en los mismos.

Ejemplo de uso del comando vgs

```
A:~# vgdisplay
  --- Volume group ---
  VG Name               vg1
  System ID
  Format                lvm2
  Metadata Areas        1
  Metadata Sequence No  1
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                0
  Open LV               0
  Max PV                0
  Cur PV                1
  Act PV                1
  VG Size               2,00 GB
  PE Size               4,00 MB
  Total PE              511
  Alloc PE / Size       0 / 0
  Free  PE / Size       511 / 2,00 GB
  VG UUID               D6QwUK-Lltf-uGg5-vH8r-ZmaK-dU0L-Lyyu3T
A:~#
```

```
A:~# vgs
  VG   #PV #LV #SN Attr   VSize VFree
  vg1    1   0   0 wz--n- 2,00G 2,00G
A:~#
```

Visualización de la información de los volúmenes lógicos

La información detallada de todos los volúmenes lógicos presentes en el sistema se muestra mediante el comando **lvdisplay**. Para obtener información más resumida, pruebe con **lvs**.

Ejemplo de uso del comando lvdisplay

```
A:~# lvdisplay
  --- Logical volume ---
  LV Name                /dev/vg1/data1
  VG Name                vg1
  LV UUID                Ll7105-aLpz-axKC-Hcuq-pPSq-QZaK-8h5PLC
  LV Write Access        read/write
  LV Status              available
  # open                 0
  LV Size                400,00 MB
  Current LE             100
  Segments               1
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     256
  Block device           253:0
A:~#
```

Ejemplo de uso del comando lvs

```
A:~# lvs
  LV    VG   Attr   LSize   Origin Snap%  Move Log Copy%  Convert
  data1 vg1  -wi-a- 400,00M
A:~#
```

c. Extensión de volúmenes lógicos

Una de las principales ventajas de los volúmenes lógicos es la manera tan sencilla de extenderlos. Hemos visto que un volumen lógico se constituye de Logical Extents proporcionados por un objeto Volume Group. Por lo tanto, es fácil extender el Logical Volume con estos Logical Extents. Dicho de otro modo, si queda espacio libre sin asignar en el grupo de volúmenes, se puede agregar a un volumen lógico ya creado. En caso contrario será necesario extender el Volume Group con anterioridad añadiendo uno o varios Physical Volumes.

Extensión de un Volume Group

La extensión de un Volume Group se realiza a partir de Physical Volume(s) con el comando**vgextend**. Los Physical Volumes se crean como se ha explicado anteriormente mediante el comando **pvcreate**.

Sintaxis del comando vgextend

```
vgextend nombre_vg dispositivo_pv
```

vgextend: opciones y parámetros	
nombre_vg	Nombre del grupo de volúmenes que se extenderá.
dispositivo_pv	Archivo especial de bloque que alberga el o los PV que nutrirán el VG.

Extensión de un Logical Volume

La extensión de un Logical Volume se realiza mediante el comando **lvextend**.

Sintaxis del comando lvextend

```
lvextend -L tamaño lv
```

lvextend: opciones y parámetros	
tamaño	Tamaño del volumen lógico que se extenderá, cuyo formato es un número seguido de la unidad. Si se antepone al tamaño el símbolo +, este tamaño se añadirá al que previamente ya tenía el volumen existente.
lv	Volumen lógico que se extenderá, identificado por su archivo especial de bloque.

Un Logical Volume no es más que un espacio de almacenamiento, independientemente del sistema de archivos que esté albergando. En el caso de que se quiera extender un Logical Volume, será necesario planear también la extensión del sistema de archivos para poder aprovechar el espacio adicional.

d. Reducción de un LV

Es posible aplicar reducciones a los elementos LVM, aunque este tipo de operaciones siempre es delicado y hay que dominarlo para su realización.

Reducción de un Logical Volume

La reducción de un volumen lógico ser realiza con el comando **lvreduce**. Los Logical Extent se retiran cuando se ejecuta el comando y todos los datos que alberga se perderán. Se deberá tomar todas las precauciones para evitar pérdidas de datos.

Reducción de un LV

```
lvreduce -L tamaño lv
```

lvreduce: opciones y parámetros	

tamaño	Tamaño que se quitará del volumen lógico extendido. Su formato es un valor numérico directamente seguido de la unidad.
lv	Volumen lógico que se reducirá, identificado mediante su archivo especial de bloque.

Reducción de un Volume Group

Un Volume Group se puede reducir con el comando **vgreduce**.

Reducción de un VG

```
vgreduce vg pv
```

vgreduce: opciones y parámetros	
vg	El grupo de volúmenes que se reducirá.
pv	El volumen físico (o los volúmenes físicos) que se retirará(n) del grupo de volúmenes.

3. Uso de volúmenes lógicos

a. Datos en los volúmenes lógicos

Una vez que se han creado los Logical Volumes, para usarlos hay que crearles un sistema de archivos. Se ha de tener bien claro que, desde un punto de vista funcional, los volúmenes lógicos son totalmente equivalentes a las particiones tradicionales creadas directamente con fdisk y de tipo Linux. El enfoque será idéntico al que se usará en particiones tradicionales, excepto el archivo especial de bloque que será el del Logical Volume.

Ejemplo de creación de un sistema de archivos ext3 en un LV

Los volúmenes lógicos aceptan la creación de sistemas de archivos como las particiones tradicionales. Observe el archivo especial de bloque bajo el que se identifica el volumen lógico.

```
A:~# mke2fs -j /dev/vg1/lv99
mke2fs 1.41.3 (12-Oct-2008)
Etiqueta del sistema de ficheros=
Tipo de SO: Linux
Tamaño del bloque=4096 (bitácora=2)
Tamaño del fragmento=4096 (bitácora=2)
Stride=0 blocks, Stripe width=0 blocks
131072 nodos-i, 524288 bloques
26214 bloques (5.00%) reservados para el superusuario
Primer bloque de datos=0
Número máximo de bloques del sistema de ficheros=536870912
16 bloque de grupos
32768 bloques por grupo, 32768 fragmentos por grupo
8192 nodos-i por grupo
Respaldo del superbloque guardado en los bloques:
   32768, 98304, 163840, 229376, 294912

Escribiendo las tablas de nodos-i: hecho
Creating journal (16384 blocks): hecho
Escribiendo superbloques y la información contable del sistema de
 ficheros: hecho

Este sistema de ficheros se revisará automáticamente cada 32 montajes
o 180 días, lo que suceda primero.  Utilice tune2fs -c o -i para
cambiarlo.
A:~#
```

De igual modo, para hacer uso de este sistema de archivos tendrá que montar el volumen lógico, ya sea de forma manual o mediante el archivo **/etc/fstab**.

Ejemplo de montaje del volumen lógico

```
A:/mnt# mount /dev/vg1/lv99 /mnt/data99
A:/mnt#
```

b. Uso de snapshot LVM para las copias de seguridad

La naturaleza flexible y evolutiva del LVM hace que sea perfectamente capaz de almacenar grandes volúmenes de datos. Sin embargo, un problema recurrente surge al generar copias de seguridad de estos volúmenes de datos. En efecto, el tiempo necesario para la copia de seguridad impide a menudo que se realicen operaciones en línea. La solución está en la funcionalidad snapshot (instantánea) disponible en las arquitecturas LVM.

Se realiza la snapshot del volumen lógico que se desea copiar cuando éste se ha montado y está en uso. Posteriormente, se efectúa la copia de seguridad a partir de la snapshot del volumen lógico que se ha generado del momento exacto en que se realizó. Hay que comprender que una snapshot no es una herramienta de copia de seguridad como tal, sino un medio que se utiliza en la estrategia de copias de seguridad.

Realización de la snapshot

Las snapshots se generan con el comando **lvcreate**. Una snapshot es por lo tanto un volumen lógico entero separado del original, que se podrá montar y usar cuando se necesite.

Hay que determinar el tamaño de la snapshot en su creación. El volumen lógico de snapshot sólo almacena físicamente las diferencias entre el volumen de producción (del que se ha generado la snapshot) y el volumen de snapshot. Si no se producen escrituras en el volumen en producción, el consumo de espacio para la snapshot será casi nulo. Si todos los datos se modifican en el volumen de producción, la snapshot explotará físicamente un espacio en disco del orden del espacio del volumen de datos en el momento en que se realizó la snapshot. El espacio usado por la snapshot puede controlarse mediante el comando **lvdisplay**.

Sintaxis del comando lvcreate para la creación de snapshots

```
lvcreate -L tamaño -s -n nombre_snapshot lv_origen
```

lvcreate para snapshots: opciones y parámetros	
-L *tamaño*	Tamaño de la snapshot que se creará.
-s	Opción que indica que se crea una snapshot de volumen lógico, y no un volumen lógico normal.
-n*nombre_snapshot*	El nombre del volumen de snapshot. Se recomienda aplicar un convenio en la nomenclatura.
lv_origen	El nombre del volumen lógico en producción a partir del cual se realizará la snapshot.

Ejemplo de creación de snapshots

La snapshot es un volumen lógico casi igual que los demás.

```
A:/mnt# lvcreate -L 1G -s -n clicclac /dev/vg1/data1
  Logical volume "clicclac" created
A:/mnt#
```

Ejemplo de visualización del espacio de disco realmente ocupado por una snapshot

En el ejemplo mostrado a continuación, no se han modificado los datos en el volumen de origen entre el lvcreate -s y el lvdisplay. Por tanto, se observa que el valor de "Allocated to snapshot" es 0%.

```
A:/mnt# lvdisplay /dev/vg1/clicclac
```

En este segundo

```
  --- Logical volume ---
  LV Name                /dev/vg1/clicclac
  VG Name                vg1
  LV UUID                xyakf0-2zMf-B3qG-S9gT-KTqw-ZJI3-W06GWi
  LV Write Access        read/write
  LV snapshot status     active destination for /dev/vg1/data1
  LV Status              available
  # open                 0
  LV Size                1,49 GB
  Current LE             381
  COW-table size         1,00 GB
  COW-table LE           256
  Allocated to snapshot  0,00%
  Snapshot chunk size    4,00 KB
  Segments               1
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     256
  Block device           253:1
A:/mnt#
```

ejemplo se han añadido datos al volumen de origen, obligando al sistema a conservar dos versiones: los datos de la snapshot, disponibles para la copia de seguridad, y los datos nuevos escritos en el disco y asignados al volumen en producción. El valor de "Allocated to snapshot" es ahora 1,45 %.

```
A:/mnt/data1# lvdisplay /dev/vg1/clicclac
  --- Logical volume ---
  LV Name                /dev/vg1/clicclac
  VG Name                vg1
  LV UUID                xyakf0-2zMf-B3qG-S9gT-KTqw-ZJI3-W06GWi
  LV Write Access        read/write
  LV snapshot status     active destination for /dev/vg1/data1
  LV Status              available
  # open                 0
  LV Size                1,49 GB
  Current LE             381
  COW-table size         1,00 GB
  COW-table LE           256
  Allocated to snapshot  1,45%
  Snapshot chunk size    4,00 KB
  Segments               1
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     256
  Block device           253:1
A:/mnt/data1#
```

Copia de seguridad de los datos de la snapshot

Desde el punto de vista del LVM, no hay que hacer nada más. Los datos están disponibles, fijados en el momento en que se creó la snapshot, y se puede generar una copia de seguridad por cualquier tipo de medio habitual.

Ejemplo de una copia de seguridad de datos de una snapshot

En este ejemplo, se monta el volumen lógico de la snapshot en un directorio /mnt/clicclac, y se crea un archivo tar comprimido de los datos que se almacenará en un dispositivo USB.

```
A:/mnt# mkdir clicclac
A:/mnt# mount /dev/vg1/clicclac clicclac
A:/mnt# ls clicclac
bigfile.tar  etc  growingfile  lost+found  midfile.tar usr
A:/mnt# tar czf /media/usb/svg_snap.tgz /mnt/clicclac
```

```
tar: Eliminando la <</>> inicial de los nombres
A:/mnt#
```

Comprobación de los conocimientos adquiridos: preguntas/respuestas

Ponga a prueba sus conocimientos respondiendo a las siguientes preguntas. Estas preguntas no siempre esperan una respuesta cerrada. Las preguntas planteadas en la certificación, a pesar de que abordan los mismos temas, en su mayoría serán planteadas en forma de preguntas de opción múltiple o que requieran una respuesta corta, en palabras escritas en el teclado.

1. Preguntas

1. ¿Por qué es necesario crear un sistema de archivos para utilizar un espacio de almacenamiento en un disco?
2. ¿Cuánto espacio de disco pueden llegar a ocupar los sistemas de archivos virtuales o pseudosistemas de archivos?
3. Los UUID sirven para identificar formalmente un sistema de archivos. ¿Qué o quién garantiza su unicidad?
4. ¿Cómo se optimiza la escritura de datos en un sistema de archivos lento?
5. ¿Por qué es difícil comprobar la coherencia del sistema de archivos raíz montado en /?
6. ¿Qué dependencia tiene el comando lsdev respecto a los pseudosistemas de archivos?
7. ¿Por qué las opciones de compresión del comando tar deben estar reservadas a las copias de seguridad en disco?
8. ¿Se requiere que el cdrom esté montado en la copia de un cdrom con el comando dd para la realización de una imagen iso?
9. ¿Cuántos discos duros se necesitan para realizar un RAID 5?
10. ¿Cuál es la diferencia entre una partición y un volumen lógico LVM?

2. Respuestas

1. ¿Por qué es necesario crear un sistema de archivos para utilizar un espacio de almacenamiento en un disco?

   Porque el sistema de archivos es el que permite organizar el espacio de almacenamiento. Sin él, una partición o un volumen lógico es tan sólo una serie de bytes sin sentido alguno. El sistema de archivos gestiona los nombres de los archivos y la ubicación física de los espacios de almacenamiento. Una cinta magnética es un ejemplo de espacio de almacenamiento sin sistema de archivos: los datos están contiguos forzosamente y no es posible modificar un archivo. Hay que borrarla y reescribirla.

2. ¿Cuánto espacio de disco pueden llegar a ocupar los sistemas de archivos virtuales o pseudosistemas de archivos?

   Ninguno. Como su nombre indica, los sistemas de archivos virtuales no existen físicamente. Habitan en memoria y se montan en un directorio del sistema de archivos real.

3. Los UUID sirven para identificar formalmente un sistema de archivos. ¿Qué o quién garantiza su unicidad?

   El azar. El UUID es, cada vez en más sistemas, la forma natural de identificar un sistema de archivos. Aunque los UUID puedan modificarse a voluntad del administrador, generalmente se asignan de forma automática en la creación de los sistemas de archivos y la aleatoriedad de 128 bits es su única garantía de unicidad.

4. ¿Cómo se optimiza la escritura de datos en un sistema de archivos lento?

   Con una escritura asíncrona: los datos escritos en disco se registran en primer lugar en memoria para más tarde escribirse en el disco. Este modo de funcionamiento, utilizado por defecto en el marco de un montaje normal, no está exento de riesgos. Los datos se consideran guardados de forma segura por parte de las aplicaciones y del usuario. En el caso de corte de la corriente eléctrica, los datos que estén en memoria pendientes de grabarse en el disco se perderán.

5. ¿Por qué es difícil comprobar la coherencia del sistema de archivos raíz montado en /?

   Porque los comandos de comprobación se ejecutan con sistemas de archivos desmontados. El sistema de

archivos raíz contiene un gran número de programas en ejecución en un sistema activo y, a menudo, los propios comandos de comprobación. Por lo tanto es imposible desmontarlo porque los programas que están en ejecución impiden la realización de esta operación. La solución consiste en forzar la comprobación en el reinicio, antes de que se monte el sistema de archivos. Se puede provocar esta comprobación de dos modos: modificando el contador de comprobación periódica con el comando e2fsck o bien forzando la comprobación con el comando shutdown y su opción -F.

6 ¿Qué dependencia tiene el comando lsdev respecto a los pseudosistemas de archivos?

El comando lsdev, como otros muchos comandos, busca la información que necesita en los archivos de los pseudosistemas de archivos (/proc/interrupts, /proc/ioports, y /proc/dma). Esto ilustra hasta qué punto es importante y útil la información que albergan los pseudosistemas de archivos.

7 ¿Por qué las opciones de compresión del comando tar deben estar reservadas a las copias de seguridad en disco?

Porque la compresión de datos hace que sean más difíciles de usar en caso de pérdida parcial. Las cintas magnéticas que se usaban tradicionalmente con el comando tar presentaban a menudo zonas de magnetización débil que las exponían a pérdidas parciales. Sin embargo, en estas circunstancias, la ausencia de compresión limitaba la pérdida a solamente los archivos albergados en estas zonas débiles.

8 ¿Se requiere que el cdrom esté montado en la copia de un cdrom con el comando dd para la realización de una imagen iso?

No. El sistema de archivos tiene que montarse si hay que copiar archivos concretos. Como el comando dd copia bloques de datos sin comprender su contenido, manipulando directamente el hardware y no los archivos, no es necesario montar el cdrom para realizar una imagen.

9 ¿Cuántos discos duros se necesitan para realizar un RAID 5?

Tres como mínimo. Dos para los datos y un tercero para la paridad. Si se incluye un disco de recambio (spare) en la configuración, este mínimo pasa a ser de cuatro: dos discos de datos, un disco de paridad y un disco preparado para reemplazar a otro disco que falle.

10 ¿Cuál es la diferencia entre una partición y un volumen lógico LVM?

Depende. Desde un punto de vista funcional ninguna: los dos tendrán un sistema de archivos y se montarán en un directorio. Sin embargo, solamente se podrá agrandar el volumen lógico cuando sea necesario (mediante el comando lvextend). No obstante, este cambio de tamaño no modificará el sistema de archivos, que deberá reorganizarse mediante un comando de redimensionamiento del sistema de archivos (resize2fs).

Trabajos prácticos

1. Uso del espacio de swap en un archivo

Se planea la instalación en la máquina alfa de una aplicación de gestión de documentación extremadamente voraz en términos de memoria principal. El presupuesto de compra de memoria adicional para el funcionamiento de esta aplicación no se desbloqueará hasta dentro de algunos meses. Se pide, por consiguiente, algún tipo de solución para que el servidor pueda soportar la carga sin quedarse colgado, aunque se degrade el rendimiento. Para ello, usted decide crear un espacio adicional de swap.

a. Creación de un archivo de swap

Comandos útiles

- cat
- chmod
- dd
- file
- mkswap
- swapon

Operaciones

1. Mostrar el swap en uso.
2. Crear en la raíz del sistema un archivo de 512 MB mediante el comando dd (si su sistema anfitrión no tiene suficiente espacio, elija un valor menor).
3. Impedir toda consulta no permitida al contenido del archivo.
4. Estructurar el archivo para que el núcleo pueda usarlo como espacio de swap.
5. Comprobar con el comando file que la operación ha sido un éxito.

Resumen de los comandos y resultado por pantalla

Mostrar el swap actual:

```
alfa:~# cat /proc/swaps
Filename                        Type    Size    Used    Priority
/dev/sda5                                       partition 369452 0 -1
alfa:~# swapon -s
Filename                        Type    Size    Used    Priority
/dev/sda5                                       partition 369452 0 -1
alfa:~#
```

Creación de un archivo de 512 MB en la raíz:

```
alfa:~# dd if=/dev/zero of=/swap bs=1024 count=524288
524288+0 records in
524288+0 records out
536970912 bytes (537 MB) copied, 2,20919 s, 243 MB/s
alfa:~#
alfa:~# file /swap
/swap: data
alfa:~# ls -lh /swap
-rw-r--r-- 1 root root 512M jul  4 22:18 /swap
alfa:~#
```

Administración de los permisos del archivo:

```
alfa:~# chmod 600 /swap
alfa:~# ls -lh /swap
-rw------- 1 root root 512M jul 31 22:18 /swap
alfa:~#
```

Estructuración del archivo:

```
alfa:~# mkswap /swap
Setting up swapspace version 1, size = 536866 kB
no label, UUID=61bbc852-9a4c-4911-9c79-323beddc6389
alfa:~#
```

Comprobación:

b.

```
alfa:~# file /swap
/swap: Linux/i386 swap file (new style), version 1 (4K pages),
size 131071 pages, no label, UUID=61bbc852-9a4c-4911-9c79-323beddc6389
alfa:~#
```

Activación del espacio de swap

Comandos útiles

- cat
- swapon

Operaciones

1. Notificar al núcleo de que debe usar este nuevo espacio de swap.
2. Comprobar que el núcleo hace uso del nuevo espacio.

Resumen de los comandos y resultado por pantalla

Activación del espacio de swap:

```
alfa:~# swapon /swap
alfa:~#
```

Comprobación con dos comandos distintos:

c. Hacer

```
alfa:~# swapon -s
Filename                        Type     Size  Used  Priority
/dev/sda5                               partition 369452 588 -1
/swap                                   file  524280 0 -2
alfa:~# cat /proc/swaps
Filename                        Type     Size  Used  Priority
/dev/sda5                               partition 369452 588 -1
/swap                                   file  524280 0 -2
alfa:~#
```

referencia en fstab

Comandos útiles

- cat
- reboot
- shutdown
- swapon
- vi

Operaciones

1. Añadir en el archivo **fstab** una línea que haga referencia al nuevo espacio de swap.
2. Reiniciar el sistema.
3. Comprobar que se haya tenido en cuenta el nuevo espacio.

Resumen de los comandos y resultado por pantalla

Archivo /etc/fstab modificado:

```
# /etc/fstab: static file system information.
#
# <file system> <mount point>   <type>  <options>       <dump>  <pass>
proc        /proc           proc    defaults          0     0
/dev/sda1   /               ext3    errors=remount-ro 0     1
/dev/sda5   none            swap    sw                0     0
/dev/sdc    /media/cdrom0   udf,iso9660 user,noauto    0     0
/dev/fd0    /media/floppy0  auto    rw,user,noauto    0     0

 # Agregar el nuevo espacio de swap

 /swap      none              swap   sw                0     0
```

Comprobación después de reiniciar:

2.

```
alfa:~# cat /proc/swaps
Filename                        Type     Size     Used    Priority
/dev/sda5                               partition 369452 0       -1
/swap                                   file      524280 0       -2
alfa:~#
```

Configuración de un disco en RAID 0

No contento con usar mucha memoria, el programa visto anteriormente necesita un espacio de almacenamiento que sea eficiente sin la obligación de que sea fiable. Claramente se ve que hay que crear un volumen lógico en RAID 0.

Añada dos discos duros virtuales SATA de 2 GB en la máquina alfa según el procedimiento visto en la introducción. Estos discos deberán ser vistos por el sistema como /dev/sdb y /dev/sdc.

a. Instalación de la gestión RAID

En el servidor alfa, instale las herramientas de gestión RAID mediante el comando siguiente:

```
apt-get install mdadm
```

Si el asistente le propone opciones de personalización, acéptelas todas por defecto.

b. Inventario de discos instalados

Comandos útiles

- dmesg
- ls

Operaciones

1. En el directorio /dev, listar todos los elementos que empiecen por hd o sd.
2. Consultar el "ring buffer" del núcleo para comprobar que los discos se han reconocido en el arranque del núcleo.
3. Identificar el disco de sistema (que tiene que estar particionado) y los dos discos añadidos.
4. Constatar la presencia en el directorio /dev de los dos archivos especiales de bloque sda y sdb.

Resumen de los comandos y resultado por pantalla

Visualización de los archivos especiales de /dev que comienzan por hd o sd:

c.

```
alfa:~# cd /dev
alfa:/dev# ls [hs]d*
sda  sda1  sda2  sda5  sdb  sdc
alfa:/dev#
```

Creación del disco RAID

Comandos útiles

- cat
- ls
- mdadm

Operaciones

1. Crear un disco RAID 0 con el nombre **md0**.
2. Comprobar la presencia del disco RAID creado por dos medios distintos.

Resumen de los comandos y resultado por pantalla

Creación del disco RAID 0:

```
alfa:/dev# mdadm -C /dev/md0 -l 0 -n 2 /dev/sdb /dev/sdc
mdadm: array /dev/md0 started.
alfa:/dev#
```

Comprobación de la presencia del disco RAID 0 por tres medios distintos:

3.

```
alfa:/dev# ls /dev/md0
/dev/md0
```

```
alfa:/dev# cat /proc/mdstat
Personalities : [raid0]
md0 : active raid0 sdc[1] sdb[0]
      4194176 blocks 64k chunks

unused devices: <none>
alfa:/dev# mdadm -D /dev/md0
/dev/md0:
        Version : 00.90
  Creation Time : Wed Sep  1 13:31:52 2010
     Raid Level : raid0
     Array Size : 4194176 (4.00 GiB 4.29 GB)
   Raid Devices : 2
  Total Devices : 2
Preferred Minor : 0
    Persistence : Superblock is persistent

    Update Time : Wed Sep  1 13:31:52 2010
          State : clean
 Active Devices : 2
Working Devices : 2
 Failed Devices : 0
  Spare Devices : 0

     Chunk Size : 64K

           UUID : 678f9e3e:f92b3780:1b3376be:99c3df95 (local to host alpha)
         Events : 0.1

    Number   Major   Minor   RaidDevice State
       0       8        0        0      active sync   /dev/sdb
       1       8       16        1      active sync   /dev/sdc
alfa:/dev#
```

Creación y uso de un volumen lógico en el disco RAID 0

Una vez creado el disco en RAID 0, desea usarlo como soporte de un volumen lógico. Esta solución es la que ofrecerá mayor flexibilidad en cuanto a futuras evoluciones del almacenamiento.

a. Instalación de las herramientas de administración de LVM

En el servidor alfa, instale las herramientas de administración LVM mediante el comando siguiente:

b.

```
alfa:/dev# apt-get install lvm2
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes extras:
  dmsetup
Se instalarán los siguientes paquetes NUEVOS:
  dmsetup lvm2
0 actualizados, 2 se instalarán, 0 para eliminar y 0 no actualizados.
Necesito descargar 393kB de archivos.
Se utilizarán 1073kB de espacio de disco adicional después de esta operación.
¿Desea continuar [S/n]?
Des:1 http://ftp.es.debian.org lenny/main lvm2 2.02.39-8 [355kB]
(...)
```

Creación del volumen lógico

Comandos útiles

- lvcreate
- lvdisplay

- pvcreate
- pvdisplay
- vgcreate
- vgdisplay

Operaciones

1. Crear un PV a partir de su disco RAID 0.
2. Comprobar el éxito del paso anterior.
3. Crear un VG llamado **volgrp** nutrido por el PV.
4. Comprobar el éxito del paso anterior.
5. Crear un LV de 1 GB llamado **documentos** a partir del VG.
6. Comprobar el éxito del paso anterior.

Resumen de los comandos y resultado por pantalla

Creación del Physical Volume a partir del disco RAID 0:

```
alfa:/dev# pvcreate /dev/md0
  Physical volume "/dev/md0" successfully created
alfa:/dev#
```

Comprobación:

```
alfa:/dev# pvdisplay
  "/dev/md0" is a new physical volume of "4,00 GB"
  --- NEW Physical volume ---
  PV Name               /dev/md0
  VG Name
  PV Size               4,00 GB
  Allocatable           NO
  PE Size (KByte)       0
  Total PE              0
  Free PE               0
  Allocated PE          0
  PV UUID               mBhGL1-i7oD-tc1k-7VX3-CQ1r-Q0AT-jAEgtj

alfa:/dev#
```

Creación del Volume Group que se alimenta del Physical Volume:

```
alfa:/dev# vgcreate volgrp /dev/md0
  Volume group "volgrp" successfully created
alfa:/dev#
```

Comprobación:

```
alfa:/dev# vgdisplay
  --- Volume group ---
  VG Name               volgrp
  System ID
  Format                lvm2
  Metadata Areas        1
```

```
  Metadata Sequence No  1
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                0
  Open LV               0
  Max PV                0
  Cur PV                1
  Act PV                1
  VG Size               4,00 GB
  PE Size               4,00 MB
  Total PE              1023
  Alloc PE / Size       0 / 0
  Free  PE / Size       1023 / 4,00 GB
  VG UUID               Dw1Qm8-BHeq-jNXN-uXVK-eaMF-gzA1-B7QwX8

alfa:/dev#
```

Creación del Logical Volume de documentos:

```
alfa:/dev# lvcreate -n documentos -L 1G volgrp
  Logical volume "documentos" created
alfa:/dev#
```

Comprobación:

c.

```
alfa:/dev# lvdisplay
  --- Logical volume ---
  LV Name                /dev/volgrp/documentos
  VG Name                volgrp
  LV UUID                xIYS6m-mq88-13br-wbp7-sp5B-iN2b-wA1GEk
  LV Write Access        read/write
  LV Status              available
  # open                 0
  LV Size                1,00 GB
  Current LE             256
  Segments               1
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     256
  Block device           253:0

alfa:/dev#
```

Creación del sistema de archivos

Comandos útiles

- mke2fs
- tune2fs

Operaciones

1. Crear un sistema de archivos de tipo ext2 en el volumen lógico.
2. A continuación, transformarlo en un sistema de archivos ext3.
3. Asignarle la etiqueta "documentos".
4. Comprobar el éxito de la operación.

Resumen de los comandos y resultado por pantalla

Creación del sistema de archivos ext2:

```
alfa:/dev# mke2fs /dev/volgrp/documentos
mke2fs 1.41.3 (12-Oct-2008)
Etiqueta del sistema de ficheros=
Tipo de SO: Linux
Tamaño del bloque=4096 (bitácora=2)
Tamaño del fragmento=4096 (bitácora=2)
65536 nodos-i, 262144 bloques
13107 bloques (5.00%) reservados para el superusuario
Primer bloque de datos=0
Número máximo de bloques del sistema de ficheros=268435456
8 bloque de grupos
32768 bloques por grupo, 32768 fragmentos por grupo
8192 nodos-i por grupo
Respaldo del superbloque guardado en los bloques:
  32768, 98304, 163840, 229376

Escribiendo tabla de nodos-i: hecho
Escribiendo superbloques y la información contable del sistema de ficheros:
hecho

Este sistema de ficheros se revisará automáticamente cada 33 montajes o
180 días, lo que suceda primero. Utilice tune2fs -c o -i para cambiarlo.
alfa:/dev#
```

Cambiamos a ext3:

```
alfa:/dev# tune2fs -j /dev/volgrp/documentos
tune2fs 1.41.3 (12-Oct-2008)
Creando el nodo-i del fichero de transacciones: hecho
Este sistema de ficheros se revisará automáticamente cada 33 montajes o
180 días, lo que suceda primero. Utilice tune2fs -c o -i para cambiarlo.
alfa:/dev#
```

Asignación de la etiqueta documentos:

```
alfa:/dev# tune2fs -L "documentos" /dev/volgrp/documentos
tune2fs 1.41.3 (12-Oct-2008)
alfa:/dev#
```

Comprobación:

d.

```
alfa:/dev# tune2fs -l /dev/volgrp/documentos | grep name
Filesystem volume name:   documentos
alfa:/dev#
```

Montaje del sistema de archivos

Comandos útiles

- cat
- mkdir
- mount

- umount

Operaciones

1. Montar su sistema de archivos en modo sólo lectura en un directorio llamado **/documentos**.
2. Comprobar el resultado.
3. Desmontarlo.
4. Añadir una línea al archivo fstab para que su sistema de archivos se monte automáticamente al arrancar el sistema.
5. Comprobar la validez de su sintaxis sin reiniciar el sistema.

Resumen de los comandos y resultado por pantalla

Creación del punto de montaje y montaje del sistema de archivos:

```
alfa:/dev# mkdir /documentos
alfa:/dev# mount -o ro /dev/volgrp/documentos /documentos
alfa:/dev#
```

Comprobación según tres métodos distintos:

```
alfa:/dev# mount
/dev/sda1 on / type ext3 (rw,errors=remount-ro)
tmpfs on /lib/init/rw type tmpfs (rw,nosuid,mode=0755)
proc on /proc type proc (rw,noexec,nosuid,nodev)
sysfs on /sys type sysfs (rw,noexec,nosuid,nodev)
udev on /dev type tmpfs (rw,mode=0755)
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev)
devpts on /dev/pts type devpts (rw,noexec,nosuid,gid=5,mode=620)
/dev/mapper/volgrp-documentos on /documentos type ext3 (ro)
alfa:/dev#
alfa:/dev# cat /proc/mounts
rootfs / rootfs rw 0 0
none /sys sysfs rw,nosuid,nodev,noexec 0 0
none /proc proc rw,nosuid,nodev,noexec 0 0
udev /dev tmpfs rw,size=10240k,mode=755 0 0
/dev/sda1 / ext3 rw,errors=remount-ro,data=ordered 0 0
tmpfs /lib/init/rw tmpfs rw,nosuid,mode=755 0 0
tmpfs /dev/shm tmpfs rw,nosuid,nodev 0 0
devpts /dev/pts devpts rw,nosuid,noexec,gid=5,mode=620 0 0
/dev/mapper/volgrp-documentos /documentos ext3 ro,errors=continue,data=
ordered 0 0
alfa:/dev#
alfa:/dev# cat /etc/mtab
/dev/sda1 / ext3 rw,errors=remount-ro 0 0
tmpfs /lib/init/rw tmpfs rw,nosuid,mode=0755 0 0
proc /proc proc rw,noexec,nosuid,nodev 0 0
sysfs /sys sysfs rw,noexec,nosuid,nodev 0 0
udev /dev tmpfs rw,mode=0755 0 0
tmpfs /dev/shm tmpfs rw,nosuid,nodev 0 0
devpts /dev/pts devpts rw,noexec,nosuid,gid=5,mode=620 0 0
/dev/mapper/volgrp-documentos /documentos ext3 ro 0 0
alfa:/dev#
```

Desmontaje del sistema de archivos:

```
alfa:/dev# umount /documentos
```

```
alfa:/dev#
```

/etc/fstab modificado:

```
# /etc/fstab: static file system information.
#
# <file system> <mount point>   <type>  <options>       <dump>  <pass>
proc        /proc          proc   defaults            0  0
/dev/sda1   /              ext3   errors=remount-ro   0  1
/dev/sda5   none           swap   sw                  0  0
/dev/sdc    /media/cdrom0  udf,iso9660 user,noauto     0  0
/dev/fd0    /media/floppy0 auto   rw,user,noauto      0  0

# Agregar el nuevo espacio de swap

/swap       none           swap   sw                  0  0

 # Montaje del volumen de documentos
 /dev/volgrp/documentos  /documentos  ext3  ro   0  0
```

Comprobación:

4.

```
alfa:/dev# mount -a

alfa:/dev# cat /proc/mounts
rootfs / rootfs rw 0 0
none /sys sysfs rw,nosuid,nodev,noexec 0 0
none /proc proc rw,nosuid,nodev,noexec 0 0
udev /dev tmpfs rw,size=10240k,mode=755 0 0
/dev/sda1 / ext3 rw,errors=remount-ro,data=ordered 0 0
tmpfs /lib/init/rw tmpfs rw,nosuid,mode=755 0 0
tmpfs /dev/shm tmpfs rw,nosuid,nodev 0 0
devpts /dev/pts devpts rw,nosuid,noexec,gid=5,mode=620 0 0
/dev/mapper/volgrp-documentos /documentos ext3
ro,errors=continue,data=ordered 0 0
alfa:/dev#
```

Ampliación del volumen lógico

Justo después de haber creado el volumen, se le comunica que el espacio de almacenamiento previsto (1 GB) se ha infradimensionado. Se necesita disponer de 3 GB. Se felicita a sí mismo por haber tomado la decisión de preferir los volúmenes lógicos a las particiones tradicionales.

a. Ampliación del LV

Comandos útiles

- df
- lvdisplay
- lvextend

Operaciones

1. Comprobar el tamaño del volumen lógico.
2. Comprobar el tamaño del sistema de archivos montado.
3. Cambiar el tamaño del volumen lógico documentos a 3 GB.
4. Comprobar el tamaño del volumen lógico.

5. Comprobar el tamaño del sistema de archivos montado.

Resumen de los comandos y resultado por pantalla

Comprobación del tamaño del volumen lógico:

```
alfa:/dev# lvdisplay /dev/volgrp/documentos
  --- Logical volume ---
  LV Name                /dev/volgrp/documentos
  VG Name                volgrp
  LV UUID                xIYS6m-mq88-13br-wbp7-sp5B-iN2b-wA1GEk
  LV Write Access        read/write
  LV Status              available
  # open                 1
  LV Size                1,00 GB
  Current LE             256
  Segments               1
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     256
  Block device           253:0
```

Comprobación del tamaño del sistema de archivos montado:

```
alfa:/dev# df -h
S. ficheros          Tamaño Usado  Disp Uso% Montado en
/dev/sda1              7,6G  1,3G  6,0G  17% /
tmpfs                   62M     0   62M   0% /lib/init/rw
udev                    10M  616K  9,4M   7% /dev
tmpfs                   62M     0   62M   0% /dev/shm
/dev/mapper/volgrp-documentos
                      1008M   34M  924M   4% /documentos
alfa:/dev#
```

Aumento del tamaño del volumen lógico a 3 GB:

```
alfa:/dev# lvextend -L 3G /dev/volgrp/documentos
  Extending logical volume documentos to 3,00 GB
  Logical volume documentos successfully resized
alfa:/dev#
```

Comprobación del tamaño del volumen lógico:

```
alfa:/dev# lvdisplay
  --- Logical volume ---
  LV Name                /dev/volgrp/documentos
  VG Name                volgrp
  LV UUID                xIYS6m-mq88-13br-wbp7-sp5B-iN2b-wA1GEk
  LV Write Access        read/write
  LV Status              available
  # open                 1
  LV Size                3,00 GB
  Current LE             768
  Segments               1
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     256
  Block device           253:0
```

```
alfa:/dev#
```

Comprobación del tamaño del sistema de archivos montado:

El

```
alfa:/dev# umount /documentos
alfa:/dev# mount /documentos
alfa:/dev#
alfa:/dev# df -h
S. ficheros          Tamaño Usado  Disp Uso% Montado en
/dev/sda1              7,6G  1,3G  6,0G  17% /
tmpfs                   62M     0   62M   0% /lib/init/rw
udev                    10M  616K  9,4M   7% /dev
tmpfs                   62M     0   62M   0% /dev/shm
/dev/mapper/volgrp-documentos
                      1008M   34M  924M   4% /documentos
alfa:/dev#
```

volumen lógico ha pasado a ser de 3 GB, pero el sistema de archivos montado, prisionero de su estructura, queda fijo a su tamaño original.

b. Ampliación del sistema de archivos

Comandos útiles

- e2fsck
- mount
- resize2fs
- umount

Operaciones

1. Desmontar el sistema de archivos.
2. Comprobar su integridad.
3. Redimensionarlo con el comando resize2fs.
4. Montarlo y comprobar el nuevo tamaño.

Resumen de los comandos y resultado por pantalla

Desmontaje del sistema de archivos y comprobación de su integridad:

```
alfa:/dev# umount /documentos
alfa:/dev# e2fsck /dev/volgrp/documentos
e2fsck 1.41.3 (12-Oct-2008)
documentos: limpio, 11/65536 archivos, 12644/262144 bloques
alfa:/dev#
```

Redimensionamiento del sistema de archivos:

```
alfa:/dev# resize2fs /dev/volgrp/documentos
resize2fs 1.41.3 (12-Oct-2008)
Resizing the filesystem on /dev/volgrp/documentos to 786432 (4k) blocks.
El sistema de ficheros en /dev/volgrp/documentos tiene ahora 786432 bloques.

alfa:/dev#
```

Montaje y comprobación del tamaño:

```
alfa:/dev# mount /documentos
alfa:/dev# df -h | grep docu
/dev/mapper/volgrp-documentos
                      3,0G   34M  2,8G   2% /documentos
alfa:/dev#
```

Requisitos y objetivos

1. Requisitos

Los conocimientos adquiridos en la certificación LPI nivel 1, especialmente:

- ☒ Visualización de los procesos y sus identificadores.
- ☒ Edición de archivos.

2. Objetivos

Cuando termine este capítulo usted será capaz de:

- ☒ Comprender el proceso de arranque de un sistema Linux.
- ☒ Comprender el uso de los niveles de ejecución.
- ☒ Gestionar el inicio de los servicios en función del nivel de ejecución.
- ☒ Conocer la existencia y la función del script rc.local.
- ☒ Cambiar de nivel de ejecución en un sistema arrancado.
- ☒ Modificar un archivo de configuración de GRUB 1.
- ☒ Añadir de forma interactiva una opción puntual al kernel en el arranque.
- ☒ Reinstalar GRUB 1 en un sistema que no funciona.
- ☒ Pasar a modo monousuario de diferentes formas.
- ☒ Configurar automáticamente GRUB 2.

El proceso init y los niveles de ejecución

1. Los niveles de ejecución

El funcionamiento de un sistema Linux se rige por los niveles de ejecución. Aunque este concepto parece un lastre hoy en día, más como una herencia del pasado que como una herramienta de administración real de una estación de trabajo Linux, su conocimiento es indispensable para una buena gestión del sistema.

Para empezar, hay que saber que un sistema Linux siempre está en algún nivel de ejecución, sea cual sea su actividad, ya se trate de un servidor apache que está respondiendo a una petición o un servidor nuevo todavía en su caja. La gestión de los niveles de ejecución consistirá en determinar cuál debe ser el comportamiento del sistema cuando éste entra en un nivel concreto.

a. ¿Qué es un nivel de ejecución?

Para simplificar, un nivel de ejecución es un nivel funcional en el que se ha determinado la lista de servicios que se arrancan o se detienen. Cuando un sistema entra en un nivel de ejecución, siempre consulta si hay que arrancar o parar servicios.

b. Los posibles niveles de ejecución

Nivel 0

El más simple: el sistema está detenido. Atención, esto no significa que no tenga que configurarse, todavía tiene que gestionar lo que sucede cuando el sistema entra en el nivel 0, es decir, cuáles son los servicios que se deben parar cuando se para físicamente una máquina.

El nivel 1 o monousuario

Un nivel un poco singular: está reservado para las operaciones de mantenimiento y solamente permite una conexión, la de la cuenta root. Además, la mayor parte de los servicios están detenidos en este nivel, lo que significa que el sistema tiene una actividad mínima. Esto es perfecto para el administrador que desea realizar operaciones de mantenimiento sin interferir con producción.

El nivel 2

En la mayoría de los sistemas, este nivel no se utiliza. Se deja a disposición del administrador que podrá establecer a partir de este nivel un modo de funcionamiento particular con solamente algunos servicios iniciados.

En los sistemas Debian y derivados (Ubuntu por ejemplo) este nivel es, en cambio, el nivel funcional por defecto.

El nivel 3

En la mayor parte de los sistemas el nivel 3 es funcional, es decir, que todos los servicios están iniciados, pero no se dispone de interfaz gráfica.

El nivel 4

En la mayoría de los sistemas este nivel no se utiliza. Se deja a disposición del administrador que podrá establecer a partir de este nivel un modo de funcionamiento particular con solamente algunos servicios iniciados.

El nivel 5

En la mayor parte de los sistemas el nivel 5 es funcional, es decir, que todos los servicios están iniciados y la interfaz gráfica está disponible.

En los sistemas Debian y derivados (Ubuntu por ejemplo) este nivel, en general, no se utiliza.

El nivel 6

Temporal por definición, el nivel 6 es el de un sistema que está reiniciando. La configuración del nivel 6 consistirá en determinar qué servicios tienen que detenerse para el reinicio del sistema. Después del reinicio se aplicará un nuevo nivel de ejecución (en general el nivel por defecto) y se iniciarán los servicios asociados a este nivel.

c. ¿Quién decide qué se encuentra en cada uno de los niveles?

En la inmensa mayoría de los casos, es la definición inicial de los niveles de ejecución la que se usa. Es decir, que el autor de la distribución (Ubuntu, Mandriva, Red Hat, etc.) elije lo que se debe activar en cada uno de los niveles de ejecución. El administrador de sistemas se conforma con esta decisión.

Sin embargo, puede darse el caso que el administrador de sistemas prefiera gestionar él mismo la configuración de los niveles de ejecución. Entonces puede elegir a qué nivel funcional corresponde cada uno de los niveles de ejecución y cuáles son los servicios asociados. A cada nivel de ejecución le corresponde entonces un conjunto de servicios.

2. Configuración del proceso init

Hemos hablado hasta ahora de niveles de ejecución como si de una lista de servicios para arrancar o parar se tratase. La cuestión ahora es saber cómo el sistema es consciente de su nivel de ejecución y de los servicios a los que se hace referencia.

a. El primer proceso iniciado en el sistema

Si se miran los procesos en ejecución en el sistema, el proceso init se encuentra en primera posición. Tiene un PPID bastante particular (0) y es el padre de bastantes otros procesos.

Los procesos

Muchos procesos tienen el número 1 como PPID.

```
[root@beta ~]# ps -ef | head
UID        PID  PPID  C STIME TTY          TIME CMD
root         1     0  0 09:07 ?        00:00:12 init [5]
root         2     1  0 09:07 ?        00:00:00 [migration/0]
root         3     1  0 09:07 ?        00:00:00 [ksoftirqd/0]
root         4     1  0 09:07 ?        00:00:00 [watchdog/0]
root         5     1  0 09:07 ?        00:00:09 [events/0]
root         6     1  0 09:07 ?        00:00:00 [khelper]
root         7     1  0 09:07 ?        00:00:00 [kthread]
root        10     7  0 09:07 ?        00:00:04 [kblockd/0]
root        11     7  0 09:07 ?        00:00:00 [kacpid]
[root@beta ~]#
```

Este proceso es el primero que se ejecuta en el arranque del kernel. Tiene evidentemente un rol privilegiado y su comportamiento se rige por un archivo de configuración: **/etc/inittab**.

b. El archivo inittab

Según las distribuciones, el archivo **/etc/inittab** tiene contenidos muy distintos, pero su estructura es siempre la misma.

Estructura del archivo /etc/inittab

```
identificador:nivel:modo_acción:comando
```

Archivo /etc/inittab: estructura de una línea de definición	

Modos de acción

identificador	Cadena alfanumérica de uno o dos caracteres. Identifica la línea. No hay más restricciones salvo evitar tener dos líneas con el mismo identificador.
nivel	El o los niveles de ejecución (en cifras) para los cuales la línea es adecuada.
modo_acción	Elegir entre varias palabras clave. Define la forma en la que el comando del cuarto campo se ejecutará.
comando	El comando que se ejecutará en el nivel o los niveles elegidos en el segundo campo según el modo de acción del tercer campo.

comunes:

- **initdefault**: siendo un poco particular, initdefault no rige la forma en la que el comando del cuarto campo se ejecutará. Además, cuando el modo de acción es initdefault, el cuarto campo está vacío. initdefault sólo sirve para definir el nivel de ejecución por defecto del sistema.
- **sysinit**: sirve para ejecutar scripts en la inicialización del sistema, independientemente del nivel de ejecución. Por esta razón sysinit no admite ningún valor en el segundo campo.
- **wait**: ejecuta el comando del cuarto campo (a menudo un script) y espera el final de esta ejecución para pasar a las siguientes líneas del archivo inittab.
- **respawn**: ejecuta el comando del cuarto campo y deja el proceso ejecutándose en segundo plano. A continuación, pasa a las siguientes líneas del archivo inittab. Si el proceso llamado por el comando se detiene, init lo volverá a iniciar automáticamente.

Archivo inittab de una distribución RedHat

Los comentarios han sido eliminados por motivos de legibilidad.

```
id:5:initdefault:
si::sysinit:/etc/rc.d/rc.sysinit
l0:0:wait:/etc/rc.d/rc 0
l1:1:wait:/etc/rc.d/rc 1
l2:2:wait:/etc/rc.d/rc 2
l3:3:wait:/etc/rc.d/rc 3
l4:4:wait:/etc/rc.d/rc 4
l5:5:wait:/etc/rc.d/rc 5
l6:6:wait:/etc/rc.d/rc 6
ca::ctrlaltdel:/sbin/shutdown -t3 -r now
pf::powerfail:/sbin/shutdown -f -h +2 "Power Failure; System Shutting Down"
pr:12345:powerokwait:/sbin/shutdown -c "Power Restored; Shutdown Cancelled"
1:2345:respawn:/sbin/mingetty tty1
2:2345:respawn:/sbin/mingetty tty2
3:2345:respawn:/sbin/mingetty tty3
4:2345:respawn:/sbin/mingetty tty4
5:2345:respawn:/sbin/mingetty tty5
6:2345:respawn:/sbin/mingetty tty6
x:5:respawn:/etc/X11/prefdm -nodaemon
```

Algunas distribuciones modernas utilizan un mecanismo de inicio alternativo (upstart) y hacen un uso limitado o nulo del archivo inittab. El principio de los niveles de ejecución y de los enlaces de ejecución de los servicios se conserva sin embargo y es idéntico en todas las distribuciones.

c. Recordatorio acerca de la ejecución de servicios

En un sistema Linux los servicios se ejecutan mediante scripts normalizados que responden como mínimo a dos condiciones:

- Se encuentran todos dentro del directorio **/etc/init.d** (o están disponibles en esta ubicación en forma de enlace simbólico).

- Admiten los parámetros **start** y **stop** para la ejecución y la parada del servicio.

Sintaxis universal de gestión de servicios

```
/etc/init.d/nombre acción
```

Gestión de servicios con el comando service

```
service nombre acción
```

Gestión de servicios: parámetros	
nombre	El nombre del servicio que se gestionará.
acción	**start** o **stop** para arrancar o parar el servicio. **status** también es una opción comúnmente soportada que indica el estado del servicio.

El comando **service**, cuando está disponible, puede considerarse como la forma recomendada de administrar servicios, ya que inicia el servicio liberándolo tanto como le es posible del entorno (pwd y variables). De este modo el servicio se inicia en un entorno más neutro.

Formato estándar de un script de gestión de un servicio

```
#!/bin/bash
case $1 in
start)
  # comando de inicio del servicio
;;
stop)
  # comando de parada del servicio
;;
esac
```

d. Enlaces entre los niveles de ejecución y los servicios

Si se examina el archivo **/etc/inittab**, hay en su interior una sección que contiene 7 líneas, una por cada nivel de ejecución, con un script como comando: **/etc/init.d/rc** en modo **wait**. No se explicará detalladamente el funcionamiento de este script, simplemente conviene recordar que inicia la ejecución de cada archivo del directorio **/etc/rc*n*.d** (siendo n el número del nivel de ejecución), con el parámetro **start** si la primera letra del nombre del archivo es una **S** y con el parámetro **stop** si la primera letra del nombre del archivo es una **K**.

Cada uno de los archivos de **/etc/rc*n*.d** es un enlace simbólico a un script de inicio de servicio de **/etc/init.d** y esta construcción permite determinar qué servicios deben iniciarse o pararse en cada uno de los niveles de ejecución.

Según la distribución, puede ser que los scripts rc y los directorios rcn.d se ubiquen en sitios distintos. La coherencia se asegura mediante la gestión correcta de las rutas en los scripts de sistema y la creación de enlaces simbólicos cuando proceda.

Estos enlaces pueden crearse manualmente con el comando **ln**, o de forma más fiable con el comando **update-rc.d**.

Creación de enlaces de gestión de servicios con el comando ln

Estos enlaces deben crearse en cada uno de los niveles de ejecución posibles.

```
cd /etc/rcx.d
ln -s ../init.d/servicio Cnnservicio
```

Enlace de ejecución de servicios: parámetros	
x	El nivel de ejecución para el que se quiere gestionar el inicio o la parada del servicio.
C	Conmutador de inicio (S) o parada (K).

nn	Número de orden en dos cifras. El script se gestionará más o menos pronto según los otros del mismo nivel.
servicio	Nombre del servicio que se gestionará.

e. Administración de los niveles de ejecución

El comando **runlevel** indica el nivel de ejecución actual.

Visualización del nivel de ejecución

```
runlevel
```

El comando **telinit** permite cambiar en caliente el nivel de ejecución de un sistema.

Cambio del nivel de ejecución

```
telinit nivel
```

Donde *nivel* representa el nivel de ejecución en el que se desea colocar el sistema.

Administración del nivel de ejecución

El cambio en caliente del nivel de ejecución sólo debería realizarse en un sistema del que se conozca la configuración.

```
alfa:~# runlevel
N 2
alfa:~# telinit 3
alfa:~#
alfa:~# runlevel
N 3
alfa:~#
```

También se puede definir el nivel de ejecución al que se desea cambiar en el kernel como parámetro en su carga. Por tanto, la elección del nivel de ejecución puede entonces realizarse desde el gestor de arranque simplemente colocando el nivel deseado en la línea de carga del kernel.

f. Comandos de gestión de enlaces de servicios

Los comandos **update-rc.d** y **chkconfig** permiten liberarse de la gestión apremiante de los enlaces de llamadas de servicios según los niveles de ejecución. Ambos comandos no están disponibles en todos los sistemas y puede ser que la creación manual de enlaces sea la única solución funcional. En todo caso, es pedagógicamente interesante comprobar la acción de estos comandos en los enlaces ubicados en los directorios **/etc/rc*n*.d**.

Creación de enlaces de gestión de servicios

```
update-rc.d servicio defaults
```

```
chkconfig --add servicio
```

Donde *servicio* representa el nombre del servicio presente en el directorio **/etc/init.d**. El parámetro defaults implica que el servicio se iniciará en los niveles funcionales por defecto y se detendrá en los niveles no funcionales (0 para la parada de sistema, 1 para el modo mantenimiento y 6 para una máquina que está reiniciando).

Supresión de enlaces de gestión de servicios

```
update-rc.d servicio remove
```

```
chkconfig --del servicio
```

Comprobación de los estados de un servicio según los niveles

```
chkconfig --list servicio
```

Ejemplo de uso del comando chkconfig

El comando chkconfig permite tanto la creación de enlaces como la visualización de los servicios según el nivel de ejecución.

Ejemplo de uso del

```
[root@beta ~]# ls /etc/rc5.d/*nfs
ls: /etc/rc5.d/*nfs: No such file or directory
[root@beta ~]# chkconfig --add nfs
[root@beta ~]# ls /etc/rc5.d/*nfs
/etc/rc5.d/K20nfs
[root@beta ~]# chkconfig --list nfs
nfs             0:off   1:off   2:off   3:off   4:off   5:off   6:off
[root@beta ~]#
```

comando update-rc.d

```
alfa:/etc/init.d# ls /etc/rc2.d/*cron
ls: no se puede acceder a /etc/rc2.d/*cron: No existe el fichero o el directorio
alfa:/etc/init.d# update-rc.d cron defaults
 Adding system startup for /etc/init.d/cron ...
   /etc/rc0.d/K20cron -> ../init.d/cron
   /etc/rc1.d/K20cron -> ../init.d/cron
   /etc/rc6.d/K20cron -> ../init.d/cron
   /etc/rc2.d/S20cron -> ../init.d/cron
   /etc/rc3.d/S20cron -> ../init.d/cron
   /etc/rc4.d/S20cron -> ../init.d/cron
   /etc/rc5.d/S20cron -> ../init.d/cron
alfa:/etc/init.d# ls /etc/rc2.d/*cron
/etc/rc2.d/S20cron
alfa:/etc/init.d#
```

g. Script independiente del nivel de ejecución: rc.local

Una vez que todos los scripts relacionados con el nivel de ejecución se han ejecutado, un último script: **rc.local** se ejecuta.

Script rc.local en una distribución ubuntu

Listo para ser usado...

```
#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.

exit 0
```

Un script /etc/rc.boot se puede encontrar en algunos sistemas antiguos. El proceso init también lo invoca.

3. Utilización de los niveles de ejecución

Sea cual sea la distribución Linux, el administrador siempre tiene a su disposición niveles de ejecución que no se utilizan por defecto. Por supuesto, no sirve de nada configurar los niveles de ejecución por diversión. En la inmensa mayoría de los casos el sistema tiene previsto un nivel funcional por defecto y todo el funcionamiento en producción se realizará en este nivel. Sin embargo, en algunos casos particulares el administrador puede elegir configurar algunos niveles por necesidades funcionales particulares y cada nivel de ejecución se corresponderá con un modo de funcionamiento del servidor, con la totalidad o parte de los servicios en funcionamiento.

Jugando con los enlaces contenidos en los directorios rc*n*.d y reemplazando la K de la primera letra por una S o viceversa se provoca, para un nivel de ejecución determinado, el arranque o la parada del servicio. De este modo, si un servicio determinado se invoca por un enlace K en el nivel 3 y un enlace S en nivel 4, el administrador podrá elegir arrancando su sistema en uno de estos dos niveles el nivel funcional del sistema.

Se puede pedir cuál es la importancia del número de orden situado detrás de la S o la K. Los scripts se tratan en el orden en el que el shell los muestra y concretamente son los caracteres alfanuméricos del nombre del enlace los que determinan el orden de ejecución o parada de los scripts. La única restricción para la asignación de este número es por lo tanto el momento en el que el script debe ejecutarse. Si un servicio depende de otro, el script que se tiene que ejecutar detrás deberá tener un número de orden superior al primero.

Los niveles de ejecución ya no se usan como herramientas de administración, las paradas de servicios se gestionan incorrectamente por defecto. Si se desea utilizar los niveles de ejecución como elementos de administración de un sistema conviene inventariar de forma precisa qué servicios se deben arrancar y cuáles detener en cada cambio.

Arranque y carga del kernel

1. El gestor de arranque GRUB

Si el proceso **init** es el primero que se ejecuta en un sistema Linux es gracias a que el kernel lo llama sistemáticamente en el arranque. Ahora sólo falta saber quién invoca al kernel: es la función del gestor de arranque.

El gestor de arranque es un pequeño programa que se encuentra generalmente en el MBR (*Master Boot Record*) y cuya función es la de provocar la carga del kernel. Para ello hay que conocer la ubicación del archivo del kernel (incluyendo su partición) y la partición que se montará en /, la raíz del sistema de archivos.

Aunque hay varios programas que cumplen esta función, GRUB (*GRand Unified Boot loader*) es el que se encuentra hoy en día en la práctica totalidad de distribuciones Linux. El gestor de arranque más extendido antes de GRUB era LILO (*LInux LOader*). LILO mostraba sus cuatro letras en el arranque y según el avance de su carga. Con ello se podía saber, en caso de error, hasta dónde había podido llegar el sistema.

GRUB hoy en día se despliega generalmente en su versión 2, pero ya que los fundamentos base de GRUB 1 (de hecho, 0.99) siguen siendo importantes, el conocimiento de ambas versiones es necesario para obtener la certificación LPI.

a. Configuración de GRUB 1

GRUB lee su configuración en el archivo **/boot/grub/menu.lst**. Para arrancar el kernel, este archivo hace referencia a algunos elementos. Según el sistema, la configuración principal puede realizarse también en el archivo **/boot/grub/grub.conf**. El archivo **menu.lst** entonces es solo un enlace a este archivo.

Formato típico de una sección de declaración de kernel en menu.lst

```
title título
root partición_kernel
kernel /ruta/kernel ro root=partición_raíz opciones
initrd/ruta/imagen_módulos
```

Archivo /boot/grub/menu.lst	
título	Como GRUB ofrece al usuario varios kernels para que elija uno de ellos, la sección título sirve para identificar el kernel que se va a cargar.
partición_kernel	La partición que alberga el kernel, en formato (hd*x,y*) donde *x*representa el número de disco duro e *y* el número de la partición. La numeración comienza con el cero.
kernel	El archivo ejecutable del kernel. Expresado en relación a la partición designada por el parámetro root.
partición_raíz	La partición que se montará en "/", expresado en formato Linux tradicional (/dev/hda1), con la etiqueta o, incluso, con el UUID.
opciones	Algunas opciones, separadas por espacios que modifican el comportamiento del kernel. Opción común: ro (read only).
imagen_módulos	El archivo imagen que permite montar un ramdisk que contenga todos los módulos del kernel que se cargarán. Expresado en relación a la partición definida por el parámetro root.

Ejemplo de menu.lst en ubuntu 10.04

Observe que los periféricos se representan con sus respectivos uuid.

```
default     0
timeout     10

title       Ubuntu 9.10, kernel 2.6.31-16-generic
kernel      /boot/vmlinuz-2.6.31-16-generic root=UUID=52200c0b-aee8-4ae0-9492
```

La directiva default en el archivo de

```
-1f488051e4a3 ro quiet splash
initrd      /boot/initrd.img-2.6.31-16-generic
quiet
```

configuración de GRUB indica el kernel que se cargará si el usuario no realiza ninguna acción. El timeout indica al cabo de cuánto tiempo cargar el kernel por defecto.

La opción ro para el montaje de la partición raíz (la que se montará en barra) permite ejecutar las herramientas de diagnóstico sin que se dañe durante la fase de inicio en caso de fallo del sistema de archivos. La opción quiet impide que el kernel sea demasiado locuaz en el arranque.

Si fuera necesario, el comando **rdev** permite comprobar cuál es la partición montada en /. Históricamente, este comando también permitía en las arquitecturas i386 parchear una imagen de kernel escribiendo los valores específicos que representaban la partición adecuada. Este comando no debería usarse salvo en última instancia.

Determinar la partición raíz con rdev

```
alfa:~# rdev
/dev/sda1 /
alfa:~#
```

b. Configuración de GRUB 2

GRUB en su versión 2 utiliza el archivo de configuración **/boot/grub/grub.cfg**. Tal y como se indica en el comienzo del archivo, no se debe modificar. Las primeras versiones de GRUB 2 publicaban este archivo incluso en modo de solo lectura. El archivo **grub.cfg** se genera automáticamente mediante el comando **update-grub**. Este comando se basa en scripts preinstalados en**/etc/grub.d** que automáticamente hacen el inventario de los sistemas arrancables. La actualización del archivo **grub.cfg** se realiza con los comandos **update-grub** o **grub-mkconfig**.

Las opciones modificables por parte del usuario (visualización del menú, tiempos de espera de arranque, etc.) se centralizan en el archivo **/etc/default/grub**. Los datos de este archivo se integrarán entonces en el archivo **grub.cfg** generado por los comandos.

Ejemplo de uso del comando update-grub

```
root@alfa:/boot/grub# update-grub
Generating grub.cfg...
Found linux image: /boot/vmlinuz-2.6.32-5-686
Found initrd image: /boot/initrd.img-2.6.32-5-686
done
root@alfa:/boot/grub#
```

Como se puede observar, el comando realiza la búsqueda de todos los elementos arrancables para integrarlos en el archivo de configuración.

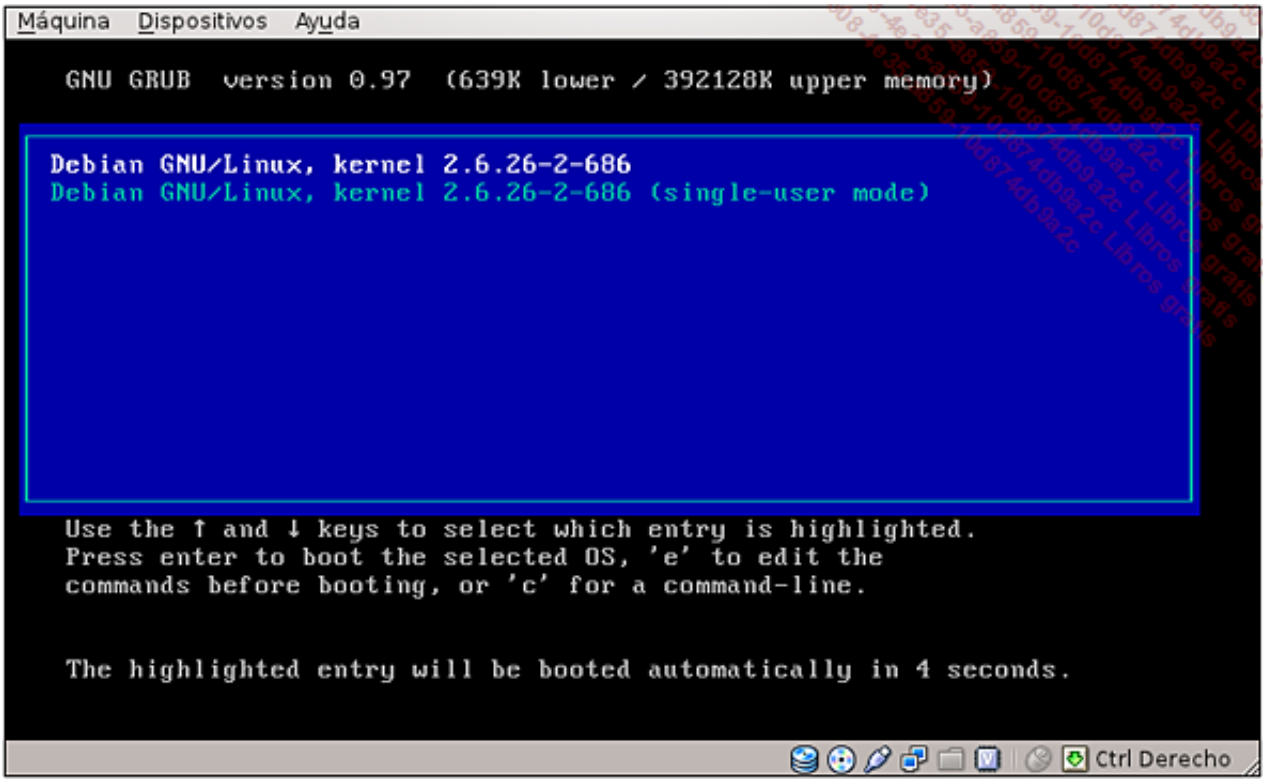
Ejemplo de uso del comando grub-mkconfig

```
root@alfa:~# grub-mkconfig -o testmenu.cfg
Generating grub.cfg...
Found linux image: /boot/vmlinuz-2.6.32-5-686
Found initrd image: /boot/initrd.img-2.6.32-5-686
done
root@alfa:~# ls
testmenu.cfg
root@alfa:~#
```

c. Funcionamiento de GRUB

GRUB propone en el arranque la carga del kernel del sistema Linux. Si coexisten varias versiones del kernel,

GRUB ofrecerá simplemente la lista de los kernels que se pueden cargar. Esta lista se muestra a partir de un conjunto de declaraciones de kernels o sistemas arrancables en el archivo **/boot/grub/menu.lst** para la versión 1 o **/boot/grub/grub.cfg** para la versión 2. Para el usuario, basta con esperar unos segundos para empezar la carga del kernel declarado por defecto en el archivo menu.lst, o bien seleccionar con las flechas de dirección y la tecla [Enter] el kernel que desea cargar.



Elección del kernel que se desea arrancar con GRUB 1

2. Utilización de GRUB 1 en modo interactivo

a. Edición de las secciones ya escritas

Si la declaración de un kernel en el archivo de configuración no se corresponde con lo que se espera (errores tipográficos en la creación del archivo, necesidades específicas...), GRUB ofrece una característica muy apreciada: la edición interactiva de las secciones ya escritas en el archivo de configuración. Para ello, basta con que en el periodo temporizado, antes de que se cargue un kernel, el usuario se posicione en el kernel que desea modificar y pulse la tecla **e**. GRUB pasará entonces al modo edición y mostrará las líneas de la sección de declaración del kernel, que se encuentran en su archivo de configuración. Se puede desplazar por cada una de estas líneas y modificarlas pulsando de nuevo sobre la tecla **e**. Cuando quede satisfecho con los cambios realizados, puede probar la carga del kernel pulsando la tecla **b** (boot). Este modo de funcionamiento representa sin lugar a dudas una de las mayores ventajas de GRUB. En efecto, es desesperante encontrarse con un sistema que no tiene medios para arrancar ni presenta posibilidad de interacción alguna.

b. Carga de un kernel no listado

Si no se dispone de entradas que se puedan modificar en GRUB (en caso de pérdida del archivo menu.lst, por ejemplo) se puede indicar directamente al gestor de arranque el conjunto de elementos necesarios. Bastará con que durante el periodo de temporizado se pulse la tecla **c** para abrir una consola interactiva.

A continuación, hay que introducir una a una todas las líneas que gestionan la carga del kernel, como si estuvieran configuradas de manera normal en el archivo **/boot/grub/menu.lst**.

Procedimiento de carga de un kernel no listado:

- Pulsar "c" durante el periodo temporizado de GRUB.
- Introducir "root (hdx,y)" donde x representa el número de disco duro e y el número de partición que alberga el kernel (la numeración empieza con el 0).

- Introducir "kernel /ruta/kernel root=partición ro quiet" donde partición es la partición que deberá montarse en "/", identificada con su archivo especial de bloque en /dev, o bien con su etiqueta o con su uuid.
- Introducir "initrd /ruta/imagen" donde imagen es el archivo de imagen de los módulos presente en principio con el archivo del kernel.
- Introducir finalmente "boot" para provocar la carga de su kernel.

Ejemplo de carga manual de un kernel:

```
c (durante el periodo temporizado, antes del arranque)
root (hd0,0)
kernel /boot/vmlinuz-2.6.26-2-686 root=/dev/hda1 ro quiet
initrd /boot/initrd.img-2.6.26-2-686
boot
```

No es necesario decir que este proceso requiere un conocimiento preciso del esquema de partición del sistema, así como los nombres de los archivos de kernel y de las imágenes. La adquisición de estos elementos no será un problema si se es capaz de arrancar de una forma u otra, pero será más complicado en caso contrario. En estas condiciones, la recuperación de estos elementos se deberá realizar a través de un tercero, como con un live cd por ejemplo.

3. Reinstalación de GRUB

a. Reinstalación simple desde un sistema activo

El comando **grub-install** reinstala GRUB en un sistema con mucha facilidad. En cambio, este método no siempre es eficaz y funciona perfectamente en caliente, justo después de un borrado accidental del gestor de arranque, por ejemplo.

Instalación de GRUB 1 con grub-install

```
grub-install --root-directory=dir_kernel disco_destino
```

grub-install: opciones y parámetros	
dir_kernel	Opcional: si el kernel no está en el sistema de archivos principal, identifica el directorio de montaje donde se encuentra el kernel.
disco_destino	El archivo especial de bloque que representa el disco en el MBR del cual se debe instalar GRUB.

Instalación de GRUB 2 con grub-install

```
grub-install dispositivo
```

Donde *dispositivo* representa el disco en el que GRUB deberá instalarse. El periférico se representa con su archivo de bloques especial.

Ejemplo de uso del comando grub-install en versión 2

```
root@alfa:~# grub-install /dev/sda
Installation finished. No error reported.
root@alfa:~#
```

b. Reinstalación desde un sistema que no arranca

La solución más fiable para reinstalar un gestor de arranque GRUB en un sistema que no puede arrancar consiste en cargar en el ordenador un live cd y realizar la reinstalación de GRUB desde el mismo. La distribución que se elija para el live cd importa poco, Knoppix o Ubuntu cumplirán muy bien con el cometido. A

continuación, después de entrar en el modo interactivo de GRUB (basta con introducir grub en un terminal), hay que definir el disco que deberá albergar el gestor de arranque y ejecutar el comando **setup** que realizará la instalación propiamente dicha del gestor.

Instalación de GRUB 1

- Desde un terminal del live cd activo, ejecute GRUB en modo interactivo tecleando "grub".
- En el shell de GRUB, defina la partición que alberga el archivo del kernel tecleando "root (hdx,y)" donde x representa el número de disco e y el número de partición (la numeración empieza con el 0).
- A continuación, introduzca "setup (hdx)" donde x representa el número de disco duro en el que GRUB debe instalarse.
- Introduzca "quit" para salir del modo interactivo de GRUB.
- Según el caso, compruebe o cree el archivo /boot/grub/menu.lst para que haga referencia correctamente al kernel o kernels que se deberán cargar.

Instalación de GRUB 2

La instalación de GRUB 2 en un sistema no arrancable se realiza mediante el comando **grub-install**desde un live-cd, como para la reinstalación en un sistema activo. La configuración automática de GRUB 2 nos facilita de gran manera la tarea.

4. Mantenimiento y modo monousuario

a. Paso a modo monousuario simplificado

El modo monousuario permite realizar operaciones de mantenimiento de un sistema. En este modo se puede conectar al sistema con la cuenta root y casi ningún servicio está en ejecución. Por tanto, el sistema está en el estado más estable posible y ninguna interacción perjudicial podrá producirse ya que el administrador trabaja solo.

Paso a modo monousuario

```
telinit 1
```

b. Apertura de una consola en caso de error en el arranque

Se puede pasar un parámetro al kernel indicándole un proceso para que lo inicie. Si este proceso se inicia en un shell, permitirá abrir una sesión interactiva, modificar archivos locales y reiniciar manualmente servicios.

Basta con editar la línea de carga del kernel en GRUB y añadir el parámetro init=/bin/bash.

Apertura de un shell directamente en el arranque

```
kernel archivo_kernel root=fs_raíz ro init=/bin/bash
```

Donde *archivo_kernel* representa el kernel que normalmente se carga y *fs_raíz* representa el sistema de archivos raíz que normalmente se carga. Solamente el parámetro init=/bin/bash debe añadirse en la línea de comandos.

Procedimiento de apertura del shell en el arranque

- Arrancar físicamente el sistema.
- Modificar la carga por defecto pulsando la tecla "e" desde la lista de sistemas disponibles.
- Añadir el parámetro init=/bin/bash al final de la línea kernel.
- Cargar el kernel pulsando la tecla "b".

La llamada directa a un shell desde el kernel permite el acceso al sistema sin tener que autentificarse. Este procedimiento demuestra que el acceso físico a una máquina sensible siempre tiene que protegerse. Se puede proteger GRUB de la edición mediante una contraseña, pero el acceso al sistema de archivos con un medio extraíble sigue siendo un peligro.

Comprobación de los conocimientos adquiridos: preguntas/respuestas

Ponga a prueba sus conocimientos respondiendo a las preguntas siguientes. Estas preguntas no siempre esperan una respuesta cerrada. Las preguntas planteadas en la certificación, a pesar de que abordan los mismos temas, en su mayoría serán planteadas en forma de preguntas de opción múltiple o que requieran una respuesta corta en palabras escritas en el teclado.

1. Preguntas

1. ¿Cuál es la diferencia entre un servicio, un daemon y un nivel de ejecución?
2. Los scripts de gestión de servicios admiten a menudo los parámetros restart y reload. ¿Cuál de estos dos parámetros consume menos recursos de sistema en su llamada?
3. ¿Qué script independiente de los niveles de ejecución se ejecuta siempre en el arranque de un sistema después de todos los scripts relacionados con el nivel de ejecución actual?
4. ¿Cuál es el resultado por pantalla del comando dmesg?
5. Para la configuración del gestor de arranque GRUB, las distribuciones inspiradas en Debian usan el archivo /boot/grub/menu.lst, mientras que los sistemas basados en Red Hat prefieren el archivo /etc/grub.conf. ¿Cómo se mantiene la coherencia para que el gestor de arranque GRUB tenga siempre acceso a su configuración en el mismo archivo?
6. ¿Cuál es la ubicación usual para poner el gestor de arranque?
7. ¿Qué interés tiene el parámetro ro (read only) generalmente pasado al kernel por el gestor de arranque que indica que la carga del kernel debe hacerse en sólo lectura?
8. ¿Qué parámetro pasado al kernel permite en su arranque acceder al sistema de forma rudimentaria en un equivalente al modo monousuario?
9. El comando telinit permite cambiar el nivel de ejecución en un sistema en funcionamiento. ¿Qué hay que hacer para que un nivel de ejecución determinado se cargue directamente en el arranque?
10. ¿Por qué la copia integral de los archivos de un disco de sistema al disco de otra máquina no basta para que funcione?

2. Respuestas

1. ¿Cuál es la diferencia entre un servicio, un daemon y un nivel de ejecución?

   Un daemon es el término inglés que representa un servicio. Un servicio es un programa residente que se ejecuta en un servidor, preparado para gestionar eventos en el sistema. Un nivel de ejecución es un estado funcional de un servidor en el que más o menos servicios deben estar en ejecución o parados. Un daemon y un servicio representan por lo tanto el mismo concepto y un nivel de ejecución describe el estado en el que se deben encontrar los servicios disponibles en el sistema.

2. Los scripts de gestión de servicios admiten a menudo los parámetros restart y reload. ¿Cuál de estos dos parámetros consume menos recursos de sistema en su llamada?

   El parámetro restart aplicado a un script de gestión de servicio ejecuta el equivalente de un stop seguido de un start. De este modo, se detienen los procesos y después se vuelven a iniciar realizando las tareas de configuración. El parámetro reload, en cambio, mantiene los procesos en ejecución, pero les obliga a que relean su configuración y se reconfiguren dinámicamente, generalmente enviando al proceso un signal 1 (hup).

3. ¿Qué script independiente de los niveles de ejecución se ejecuta siempre en el arranque de un sistema después de todos los scripts relacionados con el nivel de ejecución actual?

   El script rc.local se ejecuta automáticamente en el arranque, después de todos los scripts asociados al nivel de ejecución actual. El administrador puede incorporar en él cualquier comando para que se ejecute en el arranque independientemente del nivel de ejecución.

4. ¿Cuál es el resultado por pantalla del comando dmesg?

   El comando dmesg muestra todos los mensajes que el kernel ha podido generar desde su arranque si disponía de un terminal activo o de un archivo de registro en un sistema de archivos montado. Ante la ausencia de estos elementos (en el arranque, el kernel no tiene nada de esto a su disposición), el kernel mantiene en

memoria su registro de eventos en lo que se llama en inglés el "kernel ring buffer". El comando dmesg también devuelve su contenido por la salida estándar.

5 Para la configuración del gestor de arranque GRUB, las distribuciones inspiradas en Debian usan el archivo /boot/grub/menu.lst, mientras que los sistemas basados en Red Hat prefieren el archivo /etc/grub.conf. ¿Cómo se mantiene la coherencia para que el gestor de arranque GRUB tenga siempre acceso a su configuración en el mismo archivo?

Mediante un enlace simbólico. La historia de Unix y Linux está llena de particularidades ligadas a una distribución o un desarrollo alternativo. Para permanecer en conformidad con el uso, el pasado o las reglas POSIX, se ha generalizado el uso de enlaces simbólicos. Es mejor el uso de enlaces simbólicos, ya que un enlace duro (del inglés hard link) no es capaz de hacer referencia a un elemento externo a su sistema de archivos y no siempre los archivos favoritos y los archivos normalizados están en el mismo sistema de archivos.

6 ¿Cuál es la ubicación usual para poner el gestor de arranque?

El Master Boot Record (MBR). Una ubicación del disco situada antes de la tabla de particiones y leída en primer término por la BIOS es el sitio ideal para un gestor de arranque.

7 ¿Qué interés tiene el parámetro ro (read only) generalmente pasado al kernel por el gestor de arranque que indica que la carga del kernel debe hacerse en sólo lectura?

En caso de haber problemas en la partición que contiene el kernel, las herramientas de diagnóstico pueden ejecutarse sin dañar los elementos accedidos en sólo lectura.

8 ¿Qué parámetro pasado al kernel permite en su arranque acceder al sistema de forma rudimentaria en un equivalente al modo monousuario?

El parámetro init= permite especificar qué ejecutable debe iniciarse directamente después de la carga del kernel. Si este ejecutable es un shell, entonces el sistema arranca y ejecuta un shell independientemente de cualquier servicio. Ésta es también una forma de acceder a un sistema del que se ha perdido la contraseña.

9 El comando telinit permite cambiar el nivel de ejecución en un sistema en funcionamiento. ¿Qué hay que hacer para que un nivel de ejecución determinado se cargue directamente en el arranque?

Modificar el archivo inittab. Contiene una línea de definición del nivel de ejecución por defecto anunciada por la palabra clave initdefault.

10 ¿Por qué la copia integral de los archivos de un disco de sistema al disco de otra máquina no basta para que funcione?

Porque el kernel Linux debe invocarse por un gestor de arranque, el cual se encuentra fuera de cualquier partición de disco y, por lo tanto, no se puede copiar con herramientas de gestión de archivo ordinarias.

Trabajos prácticos

1. Creación de un nivel de ejecución personalizado con aplicaciones específicas

Aunque la gestión de los niveles personalizados ha caído en desuso en producción, esta operación constituye un muy buen ejercicio para la comprensión de los niveles de ejecución.

a. Definición de las necesidades funcionales

Han surgido nuevas necesidades en el sistema. Usted piensa que la gestión de estos servicios mediante niveles de ejecución es la mejor respuesta a estas necesidades.

Su servidor alfa debe ser funcional con sus servicios usuales iniciados en su nivel de ejecución por defecto y disponer de los mismos servicios con el añadido de una aplicación específica en ejecución en un nivel personalizado. Para poder supervisar la nueva aplicación cuando ésta entra en ejecución, va a crear una aplicación de monitorización de memoria.

Como el nivel por defecto de los servidores Debian es el nivel 2, lo conservará como nivel por defecto y personalizará el nivel 3 para que la aplicación especial (a la espera de su aplicación de monitorización) se inicie automáticamente.

b. Creación de la aplicación específica

Usted desea disponer, en un nivel de ejecución determinado, de registros periódicos del consumo de memoria en el servidor. Para ello creará el programa encargado de realizar dicho registro periódico así como su script de inicio normalizado. Cree en el directorio **/opt/scripts** con su editor favorito el archivo **supervisamem** con el contenido siguiente:

```
#!/bin/bash
while true
  do
  ahora=$(date "+%H:%M:%S - ")
  echo -n $ahora >> /var/log/supervisamem.log
  grep Dirty /proc/meminfo >> /var/log/supervisamem.log
  sleep 30
done
```

Comandos útiles

- chmod
- tail
- vi

Operaciones

1. Hacer este archivo ejecutable.
2. Crear en el directorio **/etc/init.d** un script normalizado de inicio de servicio para la aplicación supervisamem.
3. Hacer este archivo ejecutable.
4. Comprobar el buen funcionamiento del programa ejecutando el servicio correspondiente.
5. Dejar que funcione unos minutos.
6. Comprobar el contenido del archivo **/var/log/supervisamem.log**.

Resumen de los comandos y resultado por pantalla

Archivo supervisamem:

```
#!/bin/bash
while true
  do
  ahora=$(date "+%H:%M:%S - ")
  echo -n $ahora >> /var/log/supervisamem.log
  grep Dirty /proc/meminfo >> /var/log/supervisamem.log
  sleep 30
done
```

Modificación de los permisos del archivo supervisamem:

Script

```
alfa # ls -l /opt/scripts
-rw-r--r-- 1 root root 187 2010-07-13 15:31 supervisamem
alfa # chmod a+x /opt/scripts/supervisamem
alfa # ls -l /opt/scripts
-rwxr-xr-x 1 root root 187 2010-07-13 15:33 supervisamem
```

/etc/init.d/supervisamem de ejecución del servicio para gestionar la aplicación supervisamem:

```
#!/bin/bash
case $1 in
start)
  /opt/scripts/supervisamem &
  ;;
stop)
  pkill supervisamem
  ;;
esac
```

Modificación de los permisos del archivo de gestión del servicio:

Prueba del

```
alfa # ls -l /etc/init.d/supervisamem
-rw-r--r-- 1 root root 102 2010-07-13 15:37 supervisamem
alfa # chmod a+x /etc/init.d/supervisamem
alfa # ls -l /etc/init.d/supervisamem
-rwxr-xr-x 1 root root 187 2010-07-13 15:37 supervisamem
```

servicio:

c.

```
alfa # /etc/init.d/supervisamem start
alfa # tail -f /var/log/supervisamem.log
18:13:25 -Dirty :       15 kB
18:13:55 -Dirty :       228 kB
18:14:25 -Dirty :       224 kB
18:14:55 -Dirty :       65 kB
( Ctrl - C )
alfa # pgrep -l supervisamem
2203 supervisamem
alfa # /etc/init.d/supervisamem stop
alfa # pgrep -l supervisamem
alfa #
```

Modificación del nivel personalizado

Crear un nivel de funcionamiento personalizado sirve para asegurar que los servicios deseados en este nivel se invocarán correctamente en el arranque del sistema. Para ello, basta con hacer que el directorio rcn.d (donde n es el nivel de ejecución que se desea configurar) contenga un enlace cuyo nombre empiece por S (mayúscula) y

cuyo destino sea el script de servicio normalizado en /etc/init.d. El mecanismo de inicialización del sistema operativo se encargará de llamar a todos los archivos de rcn.d cuya primera letra es una S con el parámetro "start". Como estos archivos son de hecho enlaces simbólicos a los scripts de inicio de gestión de servicios y estos servicios deben responder al parámetro "start" cuando arrancan, cada enlace provocará el inicio del servicio.

Comando útil

- update-rc.d

Operaciones

1. Cree los enlaces de gestión del servicio.
2. Modifique el enlace de nivel 2 para que el servicio no arranque en modo normal.
3. Asegúrese de que el servicio arrancará correctamente en el nivel de ejecución 3.

Resumen de los comandos y resultado por pantalla

Creación de enlaces de gestión de servicio:

```
alfa:~# update-rc.d supervisamem defaults
update-rc.d: using dependency based boot sequencing
insserv: warning: script 'supervisamem' missing LSB tags and overrides
alfa:~#
```

Modificación del enlace de nivel 2:

```
alfa:~# cd /etc/rc2.d/
alfa:/etc/rc2.d# ls *supervisa*
S19supervisamem
alfa:/etc/rc2.d# mv S19supervisamem K81 supervisamem
alfa:/etc/rc2.d#
```

Verificación del nivel 3:

```
alfa:~# cd /etc/rc3.d/
alfa:/etc/rc3.d# ls *supervisa*
S19supervisamem
alfa:/etc/rc3.d#
```

d. Cambio del nivel de ejecución en caliente

Como no se ha modificado el nivel de ejecución por defecto, el sistema debe arrancar en nivel 2. Decide cambiar el nivel de ejecución en caliente para comprobar que el servicio se inicia correctamente.

Comandos útiles

- pgrep
- reboot
- runlevel
- shutdown
- telinit

Operaciones

1. Reiniciar la máquina.
2. Comprobar que la aplicación **supervisamem** no se haya iniciado en el arranque.
3. Comprobar el nivel de ejecución actual después del arranque.
4. Ejecutar un comando para que el sistema pase al nivel de ejecución 3 en caliente.
5. Comprobar que la aplicación **supervisamem** se está ejecutando.
6. Volver al nivel 2.

Resumen de los comandos y resultado por pantalla

Comprobación del nivel actual:

```
alfa:~# runlevel
N 2
alfa:~#
```

Comprobación de que no se está ejecutando la aplicación piloto (supervisamem):

```
alfa:~# pgrep -l supervisamem
alfa:~#
```

Cambio del nivel de ejecución en caliente:

```
alfa:~# telinit 3
INIT : Switching to runlevel: 3
alfa:~#
```

Comprobación del nivel actual:

```
alfa:~# runlevel
2 3
alfa:~#
```

Comprobación de que se está ejecutando la aplicación piloto (supervisamem):

```
alfa:~# pgrep -l supervisamem
2193 supervisamem
alfa:~#
```

Retorno al nivel 2:

e.

```
alfa:~# telinit 2
INIT : Switching to runlevel: 2
alfa:~#
```

Borrado de los enlaces

El responsable de la aplicación le informa de que la nueva aplicación que se despliega tiene un consumo de memoria fijo de 32 KB. Decepcionado, decide entonces eliminar los enlaces de gestión del servicio. Sin recordar el comando que permite borrar un enlace simbólico, decide ejecutar directamente un comando de gestión de servicios.

Comando útil

- update-rc.d

Operaciones

1. Eliminar todos los enlaces de los directorios **rc*n*.d** sin utilizar el comando rm (el script contenido en init.d seguirá siempre presente, ya que puede que el comando usado tenga escrúpulos. Haga lo necesario).

Resumen de los comandos y resultado por pantalla

Borrado de los enlaces de gestión del servicio:

2.

```
alpha:/etc/rc3.d# update-rc.d supervisamem remove
update-rc.d: /etc/init.d/supervisamem exists during rc.d purge (use -f to force)
alpha:/etc/rc3.d# update-rc.d -f supervisamem remove
 Removing any system startup links for
 /etc/init.d/supervisamem ...
   /etc/rc0.d/K05supervisamem
   /etc/rc1.d/K05supervisamem
   /etc/rc2.d/K05supervisamem
   /etc/rc3.d/S95supervisamem
   /etc/rc4.d/K05supervisamem
   /etc/rc5.d/K05supervisamem
   /etc/rc6.d/K05supervisamem
alpha:/etc/rc3.d#
```

Reinstalación de GRUB 1 después de haberse corrompido

Puede llegar a darse el caso de que el gestor de arranque se corrompa o se borre por accidente. Usted, un poco inquieto con esta idea, decide entrenarse reinstalando GRUB en un sistema que está desprovisto de él.

En primer lugar, copiará la totalidad de los datos de la máquina alfa (el sistema y los datos de las aplicaciones) en un disco que llamaremos clonehd. A continuación, creará una nueva máquina virtual que usará este disco pero que, naturalmente, será incapaz de arrancar. Finalmente, instalará GRUB en este disco para que pueda producirse el arranque de forma normal.

a. Creación de una máquina de test

Descargue una imagen iso de una versión anterior de Debian para instalar un sistema que use GRUB 1. Estas imágenes están disponibles en el sitio de archivos Debian:http://cdimage.debian.org/cdimage/archive. Una versión 5 o anterior será suficiente.

Cree una máquina virtual a la que llamará **deb5**. Asígnele un disco duro IDE e instale el sistema con un mínimo de recursos.

b. Copia de los datos del disco

Comandos útiles

- cp
- dmesg
- grep
- mkdir
- mke2fs
- mount

Operaciones

1. Añada el disco IDE **clonehd** a la máquina virtual deb5. Para simplificar la operación y evitar toda interacción perjudicial, elimine el controlador y los discos SATA.
2. Añada la imagen **DSL.iso** como cdrom de la máquina deb5.
3. Arranque la máquina desde el livecd DSL (la opción de arranque "dsl lang=es 2" permite tener un teclado español y un inicio sin interfaz gráfica).
4. Desde un terminal en DSL, consulte el ring-buffer del kernel para ver si el sistema ha reconocido los dos discos.
5. Cree una partición de 1 giga en el segundo disco duro.
6. Cree un sistema de archivos ext3 en esta partición.
7. Cree dos directorios **/uno** y **/dos** en el sistema de archivos del sistema virtual.
8. Monte el sistema de archivos del primer disco duro (sistema deb5) en **/uno**.
9. Monte el sistema de archivos del segundo disco duro (nuevo disco clonehd) en**/dos**.
10. Copie los datos del primer disco al segundo. Utilice una opción que conserve todos los atributos de los archivos, entre otros las fechas y los permisos.
11. Detenga la máquina virtual.

Resumen de los comandos y resultado por pantalla

Detección de discos:

```
[~]# dmesg | grep hd
<6>    ide0: BM-DMA at 0xd000-0xd007, BIOS settings: hda:DMA, hdb:DMA
<6>    ide1: BM-DMA at 0xd008-0xd00f, BIOS settings: hdc:DMA, hdd:pio
<4>hda: VBOX HARDDISK, ATA DISK drive
<4>hdb: VBOX HARDDISK, ATA DISK drive
<4>hdc: VBOX CD-ROM, ATAPI CD/DVD-ROM drive
<4>hda: attached ide-disk driver.
<6>hda: 16777216 sectors (8590 MB) w/256KiB Cache, CHS=1044/255/63
<4>hdb: attached ide-disk driver.
<6>hdb: 4194304 sectors (2147 MB) w/256KiB Cache, CHS=520/128/63
<6> hda: hda1 hda2 < hda5 >
<6> hdb: unknown partition table
<4>hdc: attached ide-scsi driver.
```

Creación de la partición:

```
[~]#   fdisk /dev/hdb
Device contains neither a valid DOS partition table, nor Sun, SGI or OSF disklabel
Building a new DOS disklabel. Changes will remain in memory only,
until you decide to write them. After that, of course, the previous
content won't be recoverable.

Warning: invalid flag 0x0000 of partition table 4 will be corrected by w(rite)

Command (m for help): p

Disk /dev/hdb: 2147 MB, 2147483648 bytes
128 heads, 63 sectors/track, 520 cylinders
Units = cylinders of 8064 * 512 = 4128768 bytes

   Device Boot    Start       End    Blocks   Id  System

Command (m for help): n
Command action
   e   extended
   p   primary partition (1-4)  p
```

```
Partition number (1-4): 1
First cylinder (1-520, default 1):
Using default value 1
Last cylinder or +size or +sizeM or +sizeK (1-520, default 520): 400

Command (m for help): p

Disk /dev/hdb: 2147 MB, 2147483648 bytes
128 heads, 63 sectors/track, 520 cylinders
Units = cylinders of 8064 * 512 = 4128768 bytes

   Device Boot    Start       End    Blocks   Id  System
/dev/hdb1               1         400     1612768+  83  Linux

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.
```

Creación del sistema de archivos en la partición.

```
[~]# mke2fs -j /dev/hdb1
mke2fs 1.34-WIP (21-May-2003)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
201760 inodes, 403192 blocks
20159 blocks (5.00%) reserved for the super user
First data block=0
13 block groups
32768 blocks per group, 32768 fragments per group
15520 inodes per group
Superblock backups stored on blocks:
  32768, 98304, 163840, 229376, 294912

Writing inode tables:  0/13  1/13  2/13  3/13  4/13  5/13  6/13
7/13  8/13  9/13 10/13 11/13 12/13 done
Creating journal (8192 blocks): done
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 22 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to override.
```

Montaje de las particiones.

```
[~]#   mkdir /uno /dos
[~]#   mount /dev/hda1 /uno
[~]#   mount /dev/hdb1 /dos
[~]#   ls /uno
bin         cdrom        dev       home        lib
media       opt          root      selinux     sys
usr         vmlinuz      root      data        etc
initrd.i    lost+found   mnt       proc        sbin
srv         tmp          var
[~]# ls /dos
lost+found
```

Copia de los datos.

Parada del

```
[~]# cp -a /uno/* /dos
[~]#   ls /dos
```

```
bin         cdrom         dev       home         lib
media       opt           root      selinux      sys
usr         vmlinuz       root      data         etc
initrd.i    lost+found    mnt       proc         sbin
srv         tmp           var
```

sistema.

c.

```
[~]# shutdown -h now
```

Creación de la máquina virtual clon

Cree una nueva máquina virtual llamada **clon**. Asígnele el disco **clonehd** que previamente habrá quitado de la máquina virtual alfa y deje todos los valores por defecto.

Iníciela y compruebe que es incapaz de arrancar aunque tenga un disco duro particionado con todos los archivos de sistema.

d. Instalación de GRUB

Comandos útiles

- grub
- grub : root
- grub : setup

Operaciones

Asignar a la máquina virtual clon la imagen iso DSL y reiniciarla. Dispondrá entonces de un shell de root en la máquina virtual.

1. Cargar la interfaz GRUB.
2. Definir la partición cuyo sistema de archivos se montará en /.
3. Instalar GRUB en el disco duro.
4. Salir de GRUB.

Resumen de los comandos y resultado por pantalla

e.

```
[~]# grub
    GRUB  version 0.91  (640K lower / 3072K upper memory)

 [ Minimal BASH-like line editing is supported.  For the first word, TAB
   lists possible command completions.  Anywhere else TAB lists the possible
   completions of a device/filename. ]

grub> root (hd0,0)
  Filesystem type is ext2fs, partittion type 0x83

grub> setup (hd0)
  Checking if "boot/grub/stage1" exists... yes
  Checking if "boot/grub/stage2" exists... yes
  Checking if "boot/grub/e2fs_stage1_5 (hd0)"... 17 sectors are embedded. succeeded
Running "install /boot/grub/stage1 d (hd0) (hd0)1+17 p
(hd(0,0)/boot/grub/stage 2 /boot/grub/menu.lst"... succeeded
Done.

grub> quit
```

Reinicio y comprobación

Desconecte la imagen iso DSL y reinicie la máquina virtual. El arranque debería producirse correctamente.

Requisitos y objetivos

1. Requisitos

Los conocimientos adquiridos con la certificación LPI de nivel 1, especialmente:

- ☒ Conocimientos generales de redes y del modelo OSI.
- ☒ Conocimientos básicos del daemon syslog.

2. Objetivos

Al final de este capítulo será capaz de:

- ☒ Configurar la red de un sistema por línea de comandos.
- ☒ Administrar rutas estáticas.
- ☒ Utilizar las herramientas de administración arp.
- ☒ Configurar los TCP Wrappers.
- ☒ Conocer los comandos de administración de redes WiFi.
- ☒ Capturar tramas en la red.
- ☒ Configurar un servidor DHCP básico.
- ☒ Configurar una reserva DHCP.
- ☒ Usar un cliente DHCP.
- ☒ Configurar IPv6.

Configuración de la red

1. Direccionamiento IP

a. Direccionamiento IPv4 y notación CIDR

Las direcciones IPv4 (direccionamiento IP histórico) se expresan en 4 bytes. Contienen dos datos fundamentales: la dirección de red y la dirección de host (la máquina que se desea identificar). Aunque el direccionamiento IP histórico ha previsto una segmentación implícita de redes según las clases A, B y C, la disociación entre la dirección de red y la dirección de host se expresa hoy en día con la máscara de subred. La máscara de subred, expresada también en 4 bytes, se compone de tantos bits a 1 como bits se usen en la dirección para describir la dirección de red en una dirección IP. El resto de bits se pone a 0.

Ejemplo de dirección IP y máscara de red asociada en decimal

192.168.1.5

255.255.255.0

Ejemplo de dirección IP y máscara de red asociada en binario

11000000.10101000.00000001.00000101

11111111.11111111.11111111.00000000

En este ejemplo, se puede observar que los 24 primeros bits de la máscara son 1, lo que indica que los 24 primeros bits de la dirección IP representan la dirección de red y los 8 restantes la dirección de host. La dirección de red es por lo tanto convencionalmente expresada como 192.168.1.0 con todos los bits de la parte host inicializados a 0.

La asociación sistemática de una máscara a las direcciones IP en las configuraciones de elementos que usan un direccionamiento IPv4 ha terminado por manifestar un problema de uso de memoria de los equipos, especialmente en los routers: conservar en memoria los 32 bits de la máscara además de los de la dirección representa una sobrecarga importante cuando se multiplica por miles de direcciones. Además, la escritura de esta máscara puede también parecer tediosa en las operaciones de configuración. Por estos motivos, ha aparecido una nueva forma de escribir la máscara, que es expresando cuántos bits a 1 tiene la máscara. La dirección IP se escribe entonces A.B.C.D/n, siendo n el número de bits a 1 de la máscara asociada a la dirección A.B.C.D. Es la notación CIDR (*Classless Internet Domain Routing*). La máscara se escribe en un solo número comprendido entre 0 y 32, y por lo tanto puede escribirse rápidamente y codificarse en un número reducido de bits en la memoria de los equipos.

Dirección IP y máscara en notación CIDR

192.168.1.5/24

b. Direccionamiento IPv6

El direccionamiento IPv6 se creó para paliar la ausencia de direcciones IPv4 anunciada desde hace ya mucho tiempo y encarar el futuro con más serenidad. El último bloque de direcciones IPv4 disponible se asignó a un operador en 2011. El paso a IPv6 deja de ser, por lo tanto, una alternativa posible y pasa a ser una realidad obligatoria a más o menos largo plazo.

Las direcciones IPv6 se expresan en 16 bytes. Aunque se tratan en binario por las máquinas, su notación para los humanos es siempre en hexadecimal, lo que requiere todavía 32 caracteres. IPv6 soporta naturalmente como su predecesor un direccionamiento jerárquico y, por lo tanto, irá sistemáticamente acompañado de una máscara de subred. Por razones comprensibles, la máscara se expresará siempre en notación CIDR.

El direccionamiento IPv6 retoma lo esencial de los principios de funcionamiento de su predecesor pero tiene algunas diferencias importantes respecto al direccionamiento IPv4.

Escritura por convenio

Como las direcciones IPv6 se expresan en 16 bytes, requieren 32 caracteres hexadecimales, con lo que son

largas y es tedioso teclearlas. Los bytes se agrupan de dos en dos y se separan por dos puntos (:). Por convenio, se puede ignorar los 0 de un grupo de dos bytes (4 caracteres hexadecimales). También se puede reemplazar cualquier serie de 0 en la dirección por dos símbolos de dos puntos (::). Naturalmente, esta facilidad puede usarse solo una vez en la escritura de la dirección.

Escritura simplificada de direcciones IPv6

20a0:1234:5678::1 es equivalente a 20a0:1234:5678:0000:0000:0000:0000:0001

20a0:1:2:3:4:5:6:7 es equivalente a 20a0:0001:0002:0003:0004:0005:0006:0007

Direccionamiento de hosts de una red IPv6

La parte host en 64 bits de una dirección IPv6 se genera localmente en la mayoría de los casos a partir de la dirección MAC de la interfaz física. La dirección MAC por lo tanto se parte en dos por la mitad, y los caracteres hexadecimales **fffe** se insertan para alcanzar los 64 bits. Por convenio, el segundo bit de menor peso del primer byte de las direcciones MAC se asigna a 1 para la creación de la dirección IPv6. La parte host de la dirección IPv6 también puede asignarse de forma pseudo-aleatoria (a partir de un hash de la dirección MAC), o simplemente configurada a mano. En este caso, se tendrá la ventaja de elegir un valor pequeño para simplificar la escritura de la dirección IPv6.

Dirección IPv6 generada a partir de la dirección MAC

```
root@alfa:~# ifconfig
eth0      Link encap:Ethernet    Hwaddr 00:1c:23:59:d1:8c
          inet adr:172.17.6.23    Bcast:172.17.6.255
Máscara:255.255.255.0
          adr inet6: fe80::21c:23ff:fe59:d18c/64 Scope:Enlace
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets 218326  bytes 267371428 (254.9 MiB)
          RX errors 0  dropped 0  overruns 0  frame 0
          TX packets 183982  bytes 24737320 (23.5 MiB)
          TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
root@alfa:~#
```

Dirección de enlace local

Cualquier máquina compatible con IPv6 dispone de una dirección llamada ”de enlace local” no enrutable y autoasignada. Esta dirección le permite comunicar con otros equipos IPv6 (servidor DHCP por ejemplo) antes de cualquier configuración más general. La dirección de enlace local comienza siempre por fe80 y la parte host de la dirección se genera localmente, en principio a partir de la dirección MAC de la interfaz.

Dirección global

Una dirección global se puede enrutar, con lo que permite a la máquina comunicarse en IPv6 con sus homólogos situados en otras redes. La parte de red de la dirección global la impone el operador o el proveedor de acceso a Internet. Naturalmente, es única en todo Internet. La parte host de la dirección se suele generar localmente a partir de la dirección MAC o se configura manualmente. La configuración de la dirección de red IPv6 puede hacerse manualmente, mediante DHCPv6 o incluso por descubrimiento automático.

Configuración automática de direcciones

En la mayoría de los casos, se usa un servidor DHCPv6 que distribuye las direcciones IPv6 a los equipos que realizan la correspondiente solicitud. El servidor se aprovechará para tener la contabilidad de las direcciones distribuidas. Las direcciones también pueden obtenerse por observación de anuncios del router, que informan sobre la dirección global usada en la red local.

Fin de broadcast

Ya raras en IPv4, las direcciones broadcast desaparecen completamente del escenario IPv6. Esto no debería ser un problema para las aplicaciones, que ya funcionan en multicast en la mayoría de los casos que requieren una comunicación de ”uno a muchos”. Los modos de comunicación son por lo tanto el unicast, el

multicast y el anycast (uno a alguien).

2. Configuración universal de la red

Cada distribución Linux intenta que la configuración de la red sea lo más fácil posible. El objetivo es a menudo el de no sufrir en comparación con Windows y hacer de algún modo que el usuario tenga a su disposición una interfaz intuitiva y fácil de configurar. Esta configuración se realiza mediante utilidades, gráficas o no, y archivos de configuración que leerán los scripts de inicio de red.

Independientemente de estos elementos de confort proporcionados por las distribuciones o los escritorios gráficos siempre se tendrá, sea cual sea la distribución y el entorno, los comandos básicos que permiten configurar la red, es decir, la dirección ip, la ruta por defecto y la dirección de los servidores DNS. El método descrito a continuación, si bien no es el más rápido, tiene la ventaja de ser universal.

a. Determinar la interfaz de red

Los sistemas Linux utilizan un nombre simbólico para la interfaz de red, ya sea real o virtual, ethernet o de otro tipo. En el caso común donde el sistema está conectado a la red mediante ethernet y sólo utiliza una tarjeta de red esta tarjeta se llamará "eth0". Se puede obtener la lista de todas las interfaces de red que existen en un sistema, configuradas o no, con el comando**ifconfig**.

Obtención de las interfaces de red con el comando ifconfig

```
ifconfig -a
```

b. Asignación de la dirección IP: ifconfig

El comando **ifconfig** tiene muchos usos y sobre todo es conocido por mostrar las direcciones MAC e IP de un sistema ya configurado. Sin embargo, el comando **ifconfig** también puede usarse para asignar dinámicamente la dirección y la máscara de red de una máquina.

Asignación de una dirección IPv4 con el comando ifconfig

```
ifconfig interfaz dirección_IPv4
ifconfig interfaz netmask máscara
```

Aunque no es el uso más común, se puede agregar una segunda dirección IP a una interfaz ya configurada.

Añadir una dirección IPv4 secundaria a una interfaz

```
ifconfig interfaz:subinterfaz dirección_IPv4
```

Añadir una dirección IPv6 con el comando ifconfig

```
ifconfig add dirección_IPv6/prefijo
```

Comando ifconfig: opciones y parámetros	
interfaz	Nombre Linux de la interfaz. Por ejemplo eth0.
subinterfaz	Nombre arbitrario de la subinterfaz. Puede ser una cadena de caracteres cualquiera.
dirección_IPv4	Dirección IPv4 que se asignará.
máscara	Valor de la máscara de subred asociada a la dirección IP.
dirección_IPv6	Dirección IPv6 que se asigna a la máquina.
prefijo	Número de bits de la máscara en notación CIDR.

c. Configuración del cliente DNS: archivo /etc/resolv.conf

Las máquinas Linux disponen de forma nativa de un cliente DNS llamado **resolver**. Toda aplicación que funcione en Linux y tenga la necesidad de hacer peticiones DNS se apoyará en este componente.

Usa el sencillo archivo de configuración **/etc/resolv.conf**, donde debe constar la referencia de al menos un servidor DNS.

Formato simplificado del archivo /etc/resolv.conf

```
search dominio
nameserver dirección_ip
```

Archivo /etc/resolv.conf: directivas y variables utilizadas	
search	Opcional: indica el sufijo de búsqueda empleado en la máquina Linux. Ahorra el tener que escribir la totalidad del nombre de dominio completamente calificado (FQDN) en las aplicaciones. El archivo /etc/resolv.conf admite varios dominios de búsqueda definidos por search.
dominio	El FQDN del dominio que constituye el sufijo de búsqueda.
nameserver	Indica la dirección IP del servidor DNS que proporcionará las resoluciones de nombre. El archivo /etc/resolv.conf admite varios servidores DNS definidos por nameserver.
dirección_ip	Dirección IP del servidor DNS al que se preguntará.

Algunas documentaciones recomiendan el uso del comando hostname -d para averiguar el sufijo DNS de un sistema. Se trata del sufijo asociado al nombre de host y no al cliente DNS. Por tanto, no se consulta en las resoluciones DNS.

d. Configuración de la puerta de enlace predeterminada: route

El comando **route** define rutas estáticas en una máquina Linux. En el marco de una configuración sencilla y puntual, se puede utilizar para definir la puerta de enlace por defecto. De hecho, consiste en declarar una ruta estática indicando la ruta por defecto.

Sintaxis del comando route para indicar una ruta estática

```
route add -net red_dest netmask máscara gw ip_p_enlace
```

Sintaxis del comando route para indicar la puerta de enlace predeterminada

```
route add -net 0.0.0.0 gw ip_p_enlace
route add default gw ip_p_enlace
```

Comando route: opciones y parámetros	
add	Indica que se añade una ruta a la tabla de enrutamiento.
-net	Indica que el destino es una red.
red_dest	La red de destino para la ruta estática que se está definiendo.
0.0.0.0	La ruta por defecto. 0.0.0.0 representa todas las redes posibles.
gw	Define el valor de la puerta de enlace predeterminada.
ip_p_enlace	Dirección IP de la puerta de enlace predeterminada que se usará.
default	Equivalente a -net 0.0.0.0.
máscara	La mácara de subred asociada a la ruta añadida.

Un servidor Linux utilizado como router soporta también los principales protocolos de enrutamiento. El software histórico **routed** soporta únicamente el protocolo de enrutamiento RIP, mientras que el software más moderno **quagga** permite el uso de casi la totalidad de protocolos de enrutamiento IP.

e. Configuración del nombre de host: hostname

El nombre de host de la máquina puede asignarse dinámicamente mediante el comando**hostname**. También permite mostrar el nombre de host del sistema si se llama sin argumentos.

Sintaxis del comando hostname para asignar un nombre de host

```
hostname nombre_host
```

nombre_host representa el nombre que se desea asignar al sistema.

Atención, este valor se conserva sólo en memoria principal y se perderá cuando se reinicie el sistema. Los sistemas tradicionales en producción deberán, por tanto, conservar este valor en un archivo de configuración para que se lea en cada arranque. Este archivo depende de la distribución. Por ejemplo, para las distribuciones basadas en Debian el archivo es **/etc/hostname**, mientras que para las distribuciones basadas en RedHat es **/etc/sysconfig/network**. Los scripts que se ejecutan en el arranque del sistema se encargan de llamar al comando **hostname** y de obtener el valor del nombre del sistema en el archivo.

Ejemplo del contenido de un archivo /etc/hostname

```
root@alfa:~$ cat /etc/hostname
alfa
```

3. Especificidad de las distribuciones

Las únicas reglas universales para la configuración de la red son las que se han descrito en los párrafos anteriores. Sin embargo, las distribuciones Linux actuales tienen procedimientos de configuración mediante scripts y archivos de configuración que se pueden clasificar en dos grandes familias: las distribuciones cuya configuración de red está en el directorio **/etc/network** y las que tienen su configuración en el directorio **/etc/sysconfig/network-scripts**.

a. Configuración de red en /etc/network

Éste es el caso de las distribuciones Debian y derivadas. Los elementos de configuración se ubican en un archivo de formato muy sencillo: /etc/network/interfaces.

Formato del archivo de configuración /etc/network/interfaces para una dirección IP estática

```
auto interfaz
iface interfaz inet static
address dirección_ip
netmask máscara
gateway ip_p_enlace
```

Formato del archivo de configuración /etc/network/interfaces para una dirección IP dinámica

```
auto interfaz
iface interfaz inet dhcp
```

Archivo interfaces: opciones y parámetros	
auto	Indica que la interfaz deberá activarse automáticamente en el arranque.
interfaz	El nombre en terminología Linux de la interfaz que se configura (ejemplo: eth0).
inet	Indica que se va a asignar una dirección IPv4.
static	Indica que la dirección IP configurada será estática.
dirección_ip	Dirección IP que se asignará a la interfaz.
máscara	Máscara de subred que se asignará a la interfaz.
ip_p_enlace	Dirección IP de la puerta de enlace por defecto.

Estos archivos no generan

dhcp	Indica que la dirección IP configurada será dinámica y se obtendrá por peticiones DHCP.

evidentemente ninguna acción por ellos mismos, el script de inicio de red (en general /etc/init.d/networking) es el que los lee e invocará al comando **ifup** (interface up) para activar las interfaces con sus respectivas configuraciones de red.

b. Configuración de red en /etc/sysconfig/network-scripts

Éste es el caso de las distribuciones RedHat y derivadas. Los elementos de configuración se ubican en un archivo de formato muy sencillo para cada interfaz en el directorio /etc/sysconfig/network-scripts. Todos estos archivos tienen el prefijo **ifcfg-** seguido del nombre de la interfaz que configuran.

Formato del archivo ifcfg-interfaz para una dirección IP estática

```
DEVICE=interfaz
BOOTPROTO=none
ONBOOT=yes
IPADDR=dirección_ip
NETMASK=máscara
GATEWAY=ip_p_enlace
```

Formato del archivo ifcfg-interfaz para una dirección IP dinámica

```
DEVICE=interfaz
BOOTPROTO=dhcp
ONBOOT=yes
```

Archivo ifcfg: opciones y parámetros	
interfaz	El nombre en terminología Linux de la interfaz que se configura (ejemplo: eth0).
BOOTPROTO=dhcp	Indica que la dirección IP configurada será dinámica y se obtendrá por peticiones DHCP.
ONBOOT=yes	Indica que la interfaz deberá activarse automáticamente en el arranque.
dirección_ip	Dirección IP que se asignará a la interfaz.
máscara	Máscara de subred que se asignará a la interfaz.
ip_p_enlace	Dirección IP de la puerta de enlace por defecto.

Sea cual sea el formato de los archivos de configuración de red, el parámetro que proporciona la dirección IP de la pasarela para la configuración de una interfaz podría hacer pensar que la puerta de enlace está asociada a la interfaz. Sin embargo, la puerta de enlace por defecto, sea el sistema que sea, es única y está asociada a la tabla de enrutamiento del sistema y no a una interfaz cualquiera.

4. Otros comandos y archivos de administración de la red

Se ha visto cómo los parámetros de red pueden configurarse mediante los comandos **ifconfig** y**route**. Sin embargo, hay muchas otras utilidades que permiten administrar, configurar y diagnosticar el funcionamiento de la red.

a. Administración de direcciones MAC con arp

Todo equipo de red que use el protocolo IP en una red ethernet tiene que utilizar el protocolo ARP para establecer la correspondencia entre direcciones IP y direcciones MAC. En un régimen de funcionamiento dinámico, el más común, una máquina que conoce la dirección IP de su destinatario pero que necesita informar su cabecera MAC para comunicarse con él envía un broadcast para solicitar si alguien en la red tiene la dirección IP en cuestión. Si la máquina destino está en el dominio broadcast (es decir, en la red local)

responderá en unicast indicando su dirección MAC. Con estas acciones se ha realizado la resolución ARP. Las correspondencias establecidas entre las direcciones MAC y las direcciones IP se conservan durante un periodo de tiempo en memoria en lo que se llama la caché ARP. En algunos casos particulares se puede asignar de forma estática una correspondencia entre una dirección IP y una dirección MAC.

El comando **arp** permite observar y eventualmente administrar los valores almacenados en esta caché.

Sintaxis del comando arp para consultar la caché

```
arp -n
```

El parámetro -n no es obligatorio, pero evita que el sistema realice una búsqueda DNS inversa que ralentiza enormemente la visualización del resultado.

Sintaxis del comando arp para borrar una entrada de la caché.

```
arp -d dirección_ip
```

Sintaxis del comando arp para asignar un valor en la caché

```
arp -s dirección_ip dirección_mac
```

Donde *dirección_ip* representa la dirección IP de la entrada que se desea administrar y*dirección_mac* representa la dirección MAC de una entrada que se asociará a una dirección IP. Las direcciones MAC se expresan en bytes en formato hexadecimal separados por dos puntos (:).

Naturalmente, el uso más habitual consiste en dejar que la totalidad de las asociaciones entre direcciones MAC y direcciones IP se realicen dinámicamente. Sin embargo si se desea configurar un gran número de asociaciones estáticas, es recomendable escribirlas en el archivo **/etc/ethers** y llamar al comando **arp** con la opción **-f**.

Formato del archivo /etc/ethers

```
dirección_mac1 dirección_ip1
dirección_mac2 dirección_ip2
...
dirección_macn dirección_ipn
```

Uso del comando arp

Se utiliza aquí el comando arp para mostrar el contenido de la caché arp antes y después de la actividad. A continuación, se asigna manualmente una dirección MAC a una dirección IP, después se tiene en cuenta el contenido del archivo /etc/ethers para configurar más asociaciones.

```
alfa:~# arp -n
alfa:~# ping 192.168.199.1
PING 192.168.199.1 (192.168.199.1) 56(84) bytes of data.
(...)
alfa:~# arp -n
Address          HWtype  HWaddress           Flags Mask  Iface
192.168.199.1    ether   08:00:27:e4:07:62   C            eth0
alfa:~# arp -s 192.168.199.222 00:01:02:a1:b2:b3
alfa:~# arp -n
Address          HWtype  HWaddress           Flags Mask  Iface
192.168.199.222  ether   00:01:02:a1:b2:b3   CM          eth0
192.168.199.1    ether   08:00:27:e4:07:62   C           eth0
alfa:~# cat /etc/ethers
00:00:00:01:02:03 192.168.199.33
00:00:00:01:02:04 192.168.199.34
00:00:00:01:02:05 192.168.199.35
00:00:00:01:02:06 192.168.199.36
alfa:~# arp -f
alfa:~# arp -n
Address          HWtype  HWaddress           Flags Mask  Iface
192.168.199.222  ether   00:01:02:a1:b2:b3   CM          eth0
192.168.199.33   ether   00:00:00:01:02:03   CM          eth0
```

```
192.168.199.35   ether   00:00:00:01:02:05   CM          eth0
192.168.199.34   ether   00:00:00:01:02:04   CM          eth0
192.168.199.36   ether   00:00:00:01:02:06   CM          eth0
192.168.199.1    ether   08:00:27:e4:07:62   C           eth0
alfa:~#
```

b. TCP Wrappers

Es posible administrar el acceso a un sistema Linux según la dirección IP o el nombre del host cliente. Se puede gestionar una lista de "todos los que están autorizados", o bien una lista de "todos los que están prohibidos". A pesar de que las modernas técnicas de intrusión y piratería informática vuelven este tipo de control de acceso casi insignificante, no deja de ser una forma de control de acceso rudimentaria que puede desalentar a curiosos. Además, la certificación LPI exige el conocimiento de estas técnicas de control de acceso.

La implementación TCPWrappers utilizada en los sistemas Linux se sustenta en la librería libwrap.

Los dos archivos que permiten este control son **/etc/hosts.allow** para los clientes autorizados, y**/etc/hosts.deny** para los clientes no autorizados. Ambos archivos son leídos por el daemon **tcpd**que aplicará los controles de acceso de forma consecuente. Por su principio de funcionamiento, estos archivos deberán usarse independientemente: si se autorizan algunos hosts para conectarse, esto significa que todo el resto de equipos tienen el acceso prohibido y por lo tanto el archivo de prohibición pierde su interés. Sin embargo, si ambos archivos están presentes en el sistema, sólo se aplicará **/etc/hosts.allow** y el archivo **/etc/hosts.deny** se ignorará.

Formato de los archivos hosts.allow y hosts.deny

```
servicio: clientes
```

TCP Wrappers: archivos de control de acceso	
servicio	El nombre del servicio cuyo acceso queremos controlar. ALL es un valor especial que representa todos los servicios posibles.
clientes	Nombre DNS o dirección IP de los clientes. Se pueden informar varios valores separados por espacios. Soporta varios comodines y formatos. ALL es un valor especial que representa todas las direcciones IP.

Ejemplo de archivo

hosts.allow

Observe que en el primer ejemplo la dirección termina con un punto. Esta sintaxis un poco particular permite incluir a todas las direcciones cuya parte prefija al punto concuerde.

```
# Todas las direcciones que empiecen por 10.1 están permitidas
ftp:10.1.
# Solamente la dirección 172.12.5.28 puede conectarse
sshd: 172.12.5.28
# Todas las direcciones de la red 192.168.1.0 pueden conectarse
ALL: 192.168.1.0/255.255.255.0
```

5. Configuración WiFi

Las distribuciones y los entornos de escritorio gráficos proporcionan utilidades gráficas para administrar las redes WiFi cuyo uso es muy intuitivo. No obstante, a continuación se mostrará cómo configurar paso a paso una conexión WiFi por línea de comandos. Las principales herramientas serán **ifconfig**, **iwconfig** e **iwlist**.

a. Determinar la interfaz WiFi

Ya sabemos que el comando **ifconfig -a** muestra todas las interfaces de red presentes en un sistema, incluso si no están activadas. Una tarjeta WiFi, por lo tanto, tiene que aparecer en la lista devuelta por este comando. Sin embargo, si el sistema tiene más tarjetas de red, un comando específico nos permitirá afinar nuestra elección.

Visualización de las interfaces WiFi con iwconfig

Todas las interfaces que devuelven una referencia a 802.11 son interfaces WiFi. En este ejemplo, es la tarjeta de red eth1.

```
usuario@ubuntu:~$ iwconfig
lo        no wireless extensions.

eth0      no wireless extensions.

eth1      IEEE 802.11  Nickname:""
          Access Point: Not-Associated
          Link Quality:5  Signal level:0  Noise level:166
          Rx invalid nwid:0  invalid crypt:0  invalid misc:0

vboxnet0  no wireless extensions.
```

b. Visualización de redes disponibles

El comando **iwlist** permite inventariar las redes disponibles.

Sintaxis del comando iwlist para la visualización de las redes del entorno

```
iwlist interfaz scan
```

Donde *interfaz* es el nombre de la tarjeta de red WiFi y scan el parámetro que indica el tipo de acción que debe realizar.

Ejemplo de scan con iwlist

En este ejemplo, se ve que hay dos redes disponibles. La primera se publica mediante un punto de acceso cuya dirección MAC es 00:0A:66:13:E7:01, usando el protocolo 802.11g (2,4 GHz et 54 Mb/s) y cuyo SSID es WLAN1. La encriptación se realiza mediante WPA-TKIP. La segunda red proviene de un punto de acceso cuya dirección MAC es CA:9D:2E:E6:B7:56, usando también el protocolo 802.11g, con el SSID hotspot y sin ningún tipo de seguridad.

```
usuario@ubuntu:~$ sudo iwlist eth1 scan
eth1      Scan completed :
          Cell 01 - Address: 00:0A:66:13:E7:01
                    ESSID:"WLAN1"
                    Mode:Managed
                    Frequency=2.437 GHz (Channel 6)
                    Quality:5/5  Signal level:-50 dBm  Noise level:-78 dBm
                    IE: WPA Version 1
                        Group Cipher : TKIP
                        Pairwise Ciphers (1) : TKIP
                        Authentication Suites (1) : PSK
                    Encryption key:on
                    Bit Rates:1 Mb/s; 2 Mb/s; 5.5 Mb/s; 11 Mb/s; 18 Mb/s
                              24 Mb/s; 36 Mb/s; 54 Mb/s; 6 Mb/s; 9 Mb/s
                              12 Mb/s; 48 Mb/s
          Cell 02 - Address: CA:9D:2E:E6:B7:56
                    ESSID:"hotspot"
                    Mode:Managed
                    Frequency:2.427 GHz (Channel 4)
                    Quality:1/5  Signal level:-83 dBm  Noise level:-84 dBm
                    Encryption key:off
                    Bit Rates:1 Mb/s; 2 Mb/s; 5.5 Mb/s; 11 Mb/s; 9 Mb/s
                              18 Mb/s; 36 Mb/s; 54 Mb/s; 6 Mb/s; 12 Mb/s
                              24 Mb/s; 48 Mb/s
```

c. Conexión a una red no segura

Una vez que se ha determinado la red, se puede conectar con el comando **iwconfig**.

Asociación a una red inalámbrica abierta

```
iwconfig interfaz essid nombre_ssid
```

Donde *interfaz* representa la interfaz de red WiFi administrada por el sistema y *nombre_ssid* el nombre de la red WiFi (*Service Set Identifier*) a la que se desea conectar.

Diagnóstico de red

1. Herramientas de diagnóstico en la capa de red

a. ping y ping6

El famoso comando **ping** sigue siendo de gran utilidad. Aparte de comprobar la conectividad IP de extremo a extremo y comprobar la resolución DNS nativa, también obtiene información más sutil, como por ejemplo la indicación de que una ruta es inaccesible.

El comando **ping6**, parecido en todas sus facetas a su homólogo histórico, permite comprobar la accesibilidad a un host remoto en IPv6.

Los comandos **ping** y **ping6** utilizan el protocolo ICMP (*Internet Control Message Protocol*).

Ejemplo de respuesta a ping

En este ejemplo, la respuesta a ping es distinta dependiendo de si la ruta existe y la máquina destino no está disponible, o si la ruta es desconocida.

```
A:~$ route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.200.0   *               255.255.255.0   U     1      0        0 eth0
A:~$ ping 172.17.18.19
connect: Network is unreachable
A:~$ route add -net 172.17.0.0 netmask 255.255.0.0 gw 192.168.200.254
A:~$ ping 172.17.18.19
PING 172.17.18.19 (172.17.18.19) 56(84) bytes of data.
From 172.17.18.19 icmp_seq=1 Destination Host Unreachable
From 172.17.18.18 icmp_seq=2 Destination Host Unreachable
A:~$
```

b. Flags del comando route

El comando **route**, utilizado para configurar las rutas estáticas, también proporciona elementos de diagnóstico. Permite saber cuáles son las redes locales o remotas (accesibles a través de una puerta de enlace) o incluso ver qué ruta es rechazada por el kernel. Esta información se obtiene a través de los flags del comando route.

Comando route: flags principales	
U	Up: la ruta está activa y es utilizable.
H	Host: el destino es un host (y no una red).
G	Gateway: el destino está accesible a través de una puerta de enlace.
D	Dynamic: la ruta ha sido configurada por un protocolo de enrutamiento.
!	El kernel ha rechazado la ruta.

Ejemplo de flags del comando route

Todas las rutas están activas y son utilizables.

```
[root@beta ~]# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.1.2.3        192.168.200.200 255.255.255.255 UGH   0      0        0 eth0
192.168.199.0   *               255.255.255.0   U     0      0        0 eth1
192.168.200.0   *               255.255.255.0   U     0      0        0 eth0
169.254.0.0     *               255.255.0.0     U     0      0        0 eth1
```

```
default         192.168.200.254 0.0.0.0         UG    0      0        0 eth0
[root@beta ~]#
```

c. traceroute

El comando **traceroute**, como el comando **ping**, permite comprobar la conectividad con un sistema remoto, pero devolviendo el conjunto de routers que permiten encaminar el paquete. En caso de problema de conectividad, se puede determinar en qué sitio se ha bloqueado o perdido el paquete.

El comando **traceroute6** permite comprobar la ruta recorrida hacia una red remota en IPv6.

Ejemplo de utilización del comando traceroute

En este ejemplo, se constata que para alcanzar la máquina 192.168.199.10 hay que pasar antes por el router 10.8.0.1.

```
usuario@ubuntu:~$ traceroute 192.168.199.10
traceroute to 192.168.199.10 (192.168.199.10), 30 hops max, 60 byte packets
 1  10.8.0.1 (10.8.0.1)  44.928 ms  50.972 ms  51.015 ms
 2  192.168.199.10 (192.168.199.10)  51.056 ms  51.112 ms  51.149 ms
usuario@ubuntu:~$
```

2. Herramientas de diagnóstico en las capas de transporte y de aplicación

a. netstat

El comando **netstat** permite observar las conexiones establecidas con el sistema local. Estas conexiones pueden ser de tipo TCP, UDP o socket. Las conexiones TCP y UDP se establecen en general con sistemas remotos, mientras que los sockets son archivos de un tipo particular que sirven de punto de intercambio entre componentes de aplicación sin acceder a la red. Por ejemplo, el servidor gráfico X que era originalmente una aplicación cliente/servidor ahora utiliza un socket para las comunicaciones entre el cliente X y su servidor situados en la misma máquina.

Con el objetivo de diagnosticar el funcionamiento de la red, son más interesantes las conexiones TCP y UDP.

Sintaxis del comando netstat para ver las conexiones activas

```
netstat -n
```

Donde la opción -n, opcional, impide la resolución inversa de las direcciones IP y de los puertos. La visualización es más rápida.

Consulta de los procesos responsables de las conexiones de red

```
netstat -p
```

Ejemplo de uso del comando netstat

A continuación se presenta el comando netstat en una sección de código en la que es llamado cada segundo para supervisar la conexión con una aplicación del sistema local. El ejemplo es un script que se compone de un bucle infinito en el que todos los comandos se han escrito en una sola línea. Si se necesita este script reiteradas veces, se deberá crear un archivo de script.

```
while true ; do clear ; netstat -an | head -20 ; sleep 1 ; done
```

Se puede salir de este bucle con la combinación de teclas [Ctrl] C.

b. nc

El comando **nc** o **netcat** es una herramienta que permite leer o escribir datos a través de una conexión de

red. Por ejemplo, si tenemos una aplicación que trabaja con conexiones TCP en el puerto 1234 y que nos está dando quebraderos de cabeza y, además, no disponemos de ninguna herramienta de diagnóstico, **nc** permite establecer una conexión a través del puerto TCP/1234, enviar datos en bruto y observar la respuesta del servidor.

Sintaxis del comando nc

```
nc -u dirección_ip puerto
```

Comando nc: opciones y parámetros	
-u	Opcional. Detalla que se desea trabajar con conexiones UDP. Si se omite, todas las peticiones se realizan usando TCP.
dirección_ip	La dirección IP de la máquina con la que se desea comunicar.
puerto	El puerto a través del cual se desea acceder a la máquina remota.

Ejemplo de uso de nc para consultar un servidor web

En este ejemplo, el servidor al que se accede responde en html al código http (GET /) que le solicita mostrar su página de inicio por defecto. Claramente se puede observar en este caso que el uso de nc para fines de diagnóstico requiere un conocimiento detallado de los protocolos subyacentes.

```
usuario@ubuntu:~$ nc 172.17.6.26 80
GET /
<html><body><h1>It works!</h1></body></html>
usuario@ubuntu:~$
```

3. Diagnóstico e información en la capa de aplicación

a. lsof

El comando **lsof** permite establecer la lista de archivos abiertos por procesos en un sistema. **lsof**, cuando se ejecuta sin opciones, muestra simplemente el conjunto de archivos que pertenecen a todos los procesos activos.

Visualización de archivos abiertos por el comando xeyes

Las columnas con más utilidad directa son PID, USER y NAME.

```
A# lsof | grep xeyes
COMMAND PID  USER  FD  TYPE DEVICE SIZE/OFF NODE    NAME
xeyes   9584 toto  cwd DIR  8,3    12288    7258113 /tmp
xeyes   9584 toto  rtd DIR  8,3     4096          2 /
xeyes   9584 toto  txt REG  8,3    20416    2366803 /usr/bin/xeyes
xeyes   9584 toto  mem REG  8,3    22568    2362738 /usr/lib/libXfixes.so.3.1.0
xeyes   9584 toto  mem REG  8,3    39232    1966225 /usr/lib/libXcursor.so.1.0.2
xeyes   9584 toto  mem REG  8,3  1170770    6463538 /usr/lib/locale/es_FR.utf8/
LC_COLLATE
xeyes   9584 toto  mem REG  8,3    19008    2146517 /lib/libuuid.so.1.3.0
xeyes   9584 toto  mem REG  8,3    22560    2364984 /usr/lib/libXdmcp.so.6.0.0
xeyes   9584 toto  mem REG  8,3    14488    2364981 /usr/lib/libXau.so.6.0.0
xeyes   9584 toto  mem REG  8,3    97904    2364446 /usr/lib/libICE.so.6.3.0
```

b. Registros en /var/log/syslog y /var/log/messages

Los archivos **/var/log/syslog** en distribuciones Debian y derivadas y **/var/log/messages** en distribuciones Red Hat y derivadas concentran lo esencial de los registros de todas las aplicaciones. Se rellenan mediante el daemon **syslogd** en el caso de Red Hat o **rsyslogd** en Debian y se incrementan con cada evento sufrido o provocado por una aplicación compatible con **syslog**. De este modo, los eventos asociados a la red, que provienen de una aplicación cliente/servidor o de la administración de la red por el sistema, se guardarán en estos archivos de registro.

En los sistemas originarios de Debian, el archivo **/var/log/daemon.log** está específicamente reservado a los registros de actividad de los servicios.

Visualización de eventos relacionados con las tarjetas de red

Los registros representan las principales fuentes de información en caso de que dejen de funcionar las aplicaciones.

```
usuario@ubuntu:/tmp$ grep eth /var/log/syslog | head
Jun 24 19:21:08 ubuntu NetworkManager: <info>  Activation (eth1)
starting connection 'Auto orange'
Jun 24 19:21:08 ubuntu NetworkManager: <info>  (eth1): device state change: 3 -> 4
(reason 0)
Jun 24 19:21:08 ubuntu NetworkManager: <info>  Activation (eth1) Stage 1 of 5
(Device Prepare) scheduled...
Jun 24 19:21:08 ubuntu NetworkManager: <info>  Activation (eth1) Stage 1 of 5
(Device Prepare) started...
Jun 24 19:21:08 ubuntu NetworkManager: <info>  Activation (eth1) Stage 2 of 5
(Device Configure) scheduled...
Jun 24 19:21:08 ubuntu NetworkManager: <info>  Activation (eth1) Stage 1 of 5
(Device Prepare) complete.
Jun 24 19:21:08 ubuntu NetworkManager: <info>  Activation (eth1) Stage 2 of 5
(Device Configure) starting...
Jun 24 19:21:08 ubuntu NetworkManager: <info>  (eth1): device state change: 4 -> 5
(reason 0)
Jun 24 19:21:08 ubuntu NetworkManager: <info>  Activation (eth1/wireless):
connection 'Auto orange' requires no security.  No secrets needed.
Jun 24 19:21:08 ubuntu NetworkManager: <info>  Activation (eth1) Stage 2 of 5
(Device Configure) complete.
```

4. libpcap y las capturas de paquetes

a. La librería libpcap

Para obtener información precisa sobre el funcionamiento a nivel de red de una aplicación, puede darse el caso de que se tenga que llegar a capturar directamente el conjunto de elementos que circulan por la red. Hay muchas herramientas para cumplir esta tarea en todos los sistemas. En entornos Linux, la mayor parte de estas herramientas se basan en la librería **libpcap** que proporciona una interfaz de bajo nivel estandarizada para la captura de paquetes. **libpcap** se creó a partir de los primeros desarrollos de un comando de captura llamado **tcpdump**. Posteriormente, fue utilizada por muchos programas de análisis de red, entre los cuales el célebre **wireshark**.

b. tcpdump

tcpdump es una herramienta que devuelve por la salida estándar (la pantalla) un resumen de las capturas realizadas por la tarjeta de red. **tcpdump** trabajando en tiempo real (durante la ejecución del programa) es útil para supervisar directamente la actividad de la red de una máquina. Si se redirigen las capturas a un archivo se conservará la información completa de los paquetes para poder usarse con otras utilidades compatibles con el formato **libpcap**.

Sintaxis del comando tcpdump

```
tcpdump -w archivo -i interfaz -s ventana -n filtro
```

tcpdump: opciones y parámetros	
-w*archivo*	Opcional: para guardar el resultado de la captura en un archivo en formato libpcap.
-i *interfaz*	Opcional: para realizar la captura desde una interfaz concreta.
-s*ventana*	Opcional: para limitar el tamaño de las tramas capturadas. Usado sobre todo con el parámetro 0 (sin límites).
-n	Opcional: no reemplazar los valores numéricos por expresiones literales.

Ejemplo de uso de tcpdump

A

filtro	Determina el tráfico que se capturará. Palabras clave: host, port, src, dest.

continuación, el ejemplo siguiente muestra elementos de tráfico de red capturados sobre la marcha por tcpdump. Observe que la brevedad de la información mostrada (en este caso, intercambios relacionados con el Spanning Tree Protocol entre conmutadores) no permite realizar un análisis en profundidad, sino principalmente ver de primera mano la naturaleza de la información intercambiada.

```
root@servidor:~$ tcpdump
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth6, link-type EN10MB (Ethernet), capture size 96 bytes
10:07:59.961927
10:08:00.019503 STP 802.1d, Config, Flags [none], bridge-id
8007.00:25:46:b4:3c:80.800c, length 43
10:08:02.034712 STP 802.1d, Config, Flags [none], bridge-id
8007.00:25:46:b4:3c:80.800c, length 43
^C
3 packets captured
3 packets received by filter
0 packets dropped by kernel
root@servidor:~$
```

Este ejemplo más detallado redirige a un archivo en formato libpcap las peticiones http a un servidor con dirección IP 192.168.50.24.

```
root@servidor:~$ tcpdump -w archivo.cap -i eth0 -s 0 -n port 80 and host 192.168.50.24
root@servidor:~$
```

c. Wireshark

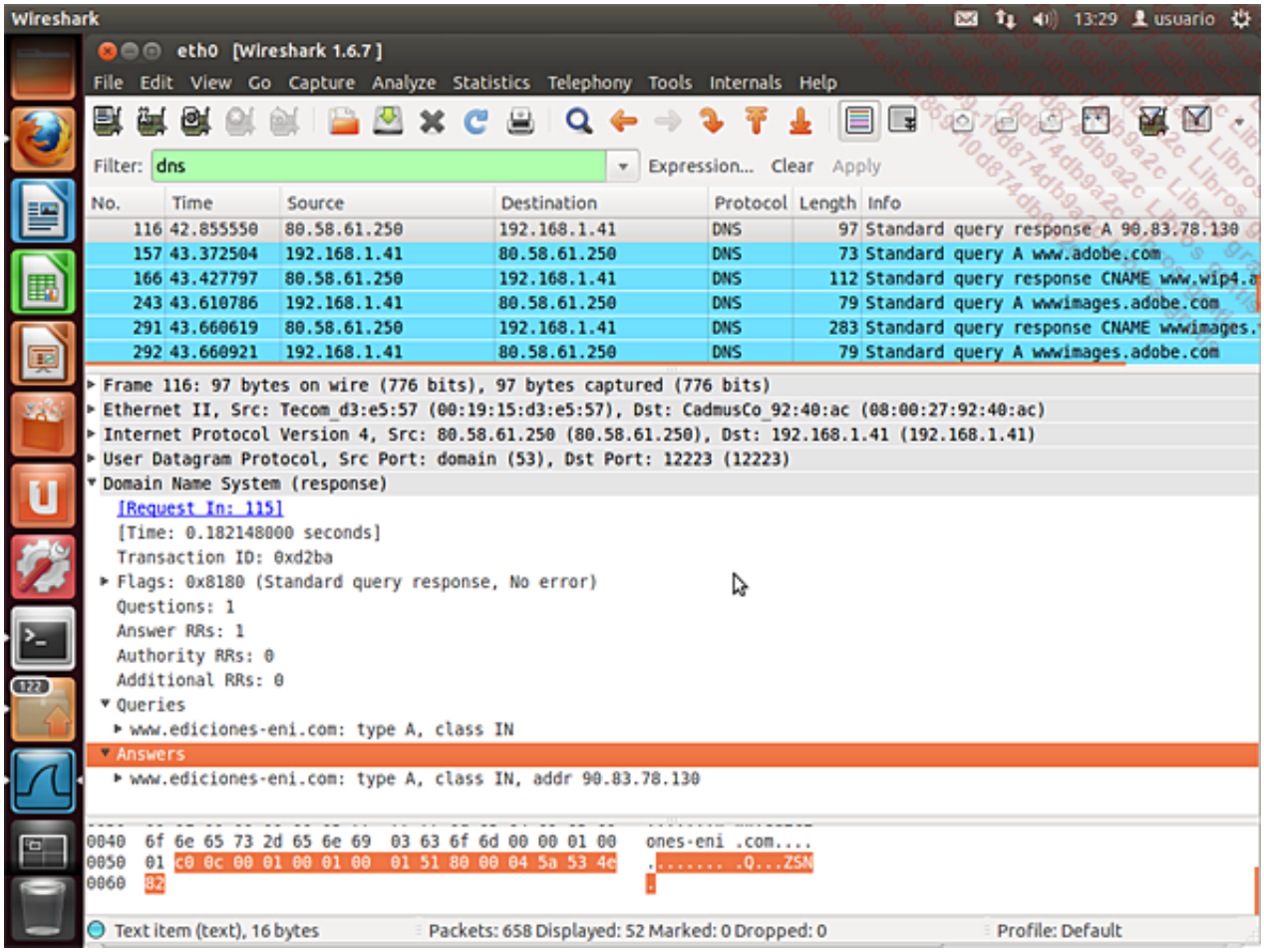
Wireshark (antiguamente **ethereal**) es una aplicación de captura de tramas multiplataforma disponible especialmente en entornos Windows y Linux. **Wireshark** se basa en la librería **libpcap** y permite guardar los datos capturados en este formato o usar las capturas realizadas por otras herramientas. **Wireshark** ofrece para cada captura un desacople según las capas del modelo OSI de los datos capturados, lo que resulta a la vez práctico y muy educativo.

Procedimiento estándar de captura con wireshark

- Ejecutar la aplicación Wireshark.
- En el menú **Capture**, elegir **Interfaces**.
- Obtener la tarjeta a la que está asociada su dirección IP.
- Hacer clic en **Start** para iniciar la captura.
- Visualizar los paquetes que se están capturando.
- Detener la captura haciendo clic en **Stop** en el menú **Capture**.

Ejemplo de captura de paquetes con wireshark

Observe que la pantalla está dividida horizontalmente en tres paneles: el paquete que se está analizando, los detalles capa por capa y el valor hexadecimal de los datos capturados. En este caso, se puede apreciar que se trata de una petición DNS de tipo A para la resolución del nombre start.ubuntu.com.



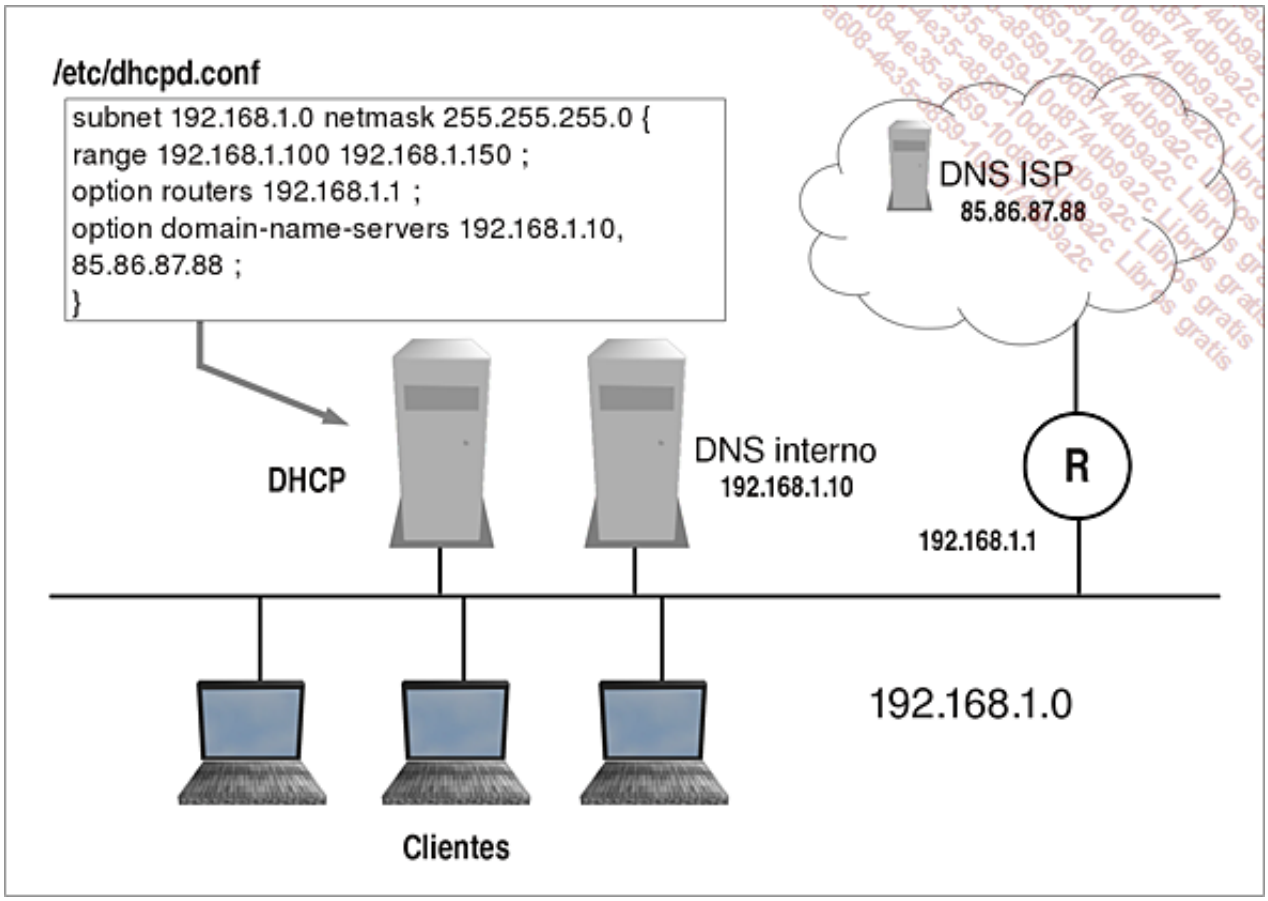
En una red congestionada, se corre el riesgo de quedar sepultado bajo una avalancha de paquetes capturados que no tienen necesariamente relación con lo que se está buscando. Se mejorará la visibilidad aplicando un filtro de visualización (campo **Filter** de la pantalla principal). Esta operación tiene la ventaja añadida de ser reversible (botón **Clear**).

Configuración automática con DHCP

1. El protocolo DHCP

DHCP (*Dynamic Host Configuration Protocol*) es un protocolo cliente/servidor que tiene como objetivo asignar automáticamente una dirección IP así como los parámetros funcionales a los equipos de la red. Todo equipo que no pueda configurarse estáticamente por un administrador de red usará este protocolo.

a. Funcionamiento



Descubrimiento de un servidor

Los clientes DHCP generan peticiones que envían por la red con la esperanza de encontrar un servidor DHCP. Esta petición inicial sólo puede enviarse en broadcast: el equipo origen que genera la petición no sabe ni siquiera su propia dirección, entonces es poco probable que conozca de antemano la dirección de un servidor DHCP.

Los paquetes enviados para el descubrimiento se llaman de forma estándar **DHCPDISCOVER**.

Primera respuesta del servidor

Si un servidor DHCP en la red recibe la petición de un cliente, le propondrá una dirección y una configuración de red. Como el cliente al que responde el servidor de direcciones no tiene todavía dirección IP, esta respuesta también se realizará en broadcast.

Los paquetes enviados para la respuesta del servidor tienen como nombre estandarizado**DHCPOFFER**.

Aceptación de la propuesta

El cliente DHCP satisfecho de la propuesta que se le ha hecho, la va a aceptar. En este estado, esta respuesta podría enviarse en unicast, debido a que el cliente ya tiene una propuesta de dirección IP y conoce la del servidor. Sin embargo, este envío todavía se realizará en broadcast. En efecto: si un segundo servidor DHCP trabaja concurrentemente con el primero para otorgar direcciones, este broadcast de aceptación

enviado a un servidor pero recibido por los dos ofertantes sirve para que el descartado tome nota de que el equipo solicitante ya ha conseguido dirección por otro camino.

Los paquetes enviados para la aceptación de la oferta del servidor se llaman **DHCPREQUEST**.

Acuse de recibo del servidor

Finalmente, el servidor realiza la asignación de la dirección y cierra la transacción enviando un acuse de recibo. El servidor aprovecha este último envío para anunciar la duración de la asignación de la dirección. Se llama a esta duración tiempo de concesión DHCP y su duración varía según la configuración del servidor entre unas horas y unos días.

Los paquetes que se envían para el acuse de recibo se llaman **DHCPACK**.

b. El servicio DHCP en sistemas Linux

El servicio DHCP más extendido en los sistemas Linux, que precisamente es el que hay que conocer para la certificación LPI, es el servicio DHCP del ISC (*Internet System Consortium*). El ISC es un organismo creado en 1994 para asegurar el desarrollo y la sostenibilidad del servidor DNS BIND, cuyo desarrollo tiene origen en la universidad de Berkeley. ISC DHCP es un desarrollo original del ISC para proporcionar una implementación de referencia de este protocolo.

El servicio se inicia mediante un script estándar en **/etc/init.d**. Su nombre varía en función de la distribución y la implementación.

2. Configuración del servidor

Los aspectos más básicos de la configuración de un servidor DHCP ISC se encuentran en el archivo **/etc/dhcpd.conf**.

En él se pueden encontrar las directivas de funcionamiento, las opciones generales del servidor y la declaración de los recursos que se asignarán. Cada línea deberá terminar con un punto y coma.

a. Funcionamiento general del servidor

Directivas principales de comportamiento del servidor en dhcpd.conf.

```
default-lease-time duración;
authoritative;
log-facility destino;
```

dhcpd.conf: comportamiento del servidor	
default-lease-time*duración*	Indica la duración de la concesión DHCP en segundos.
authoritative	Opcional. Todo cliente que solicite la renovación de una dirección fuera de rango deberá renunciar a ella.
log-facility *destino*	Administración de los registros: envía los eventos a la "facility" *destino* del servidor syslog.

b. Parámetros transmitidos a los clientes

En el archivo de configuración se pueden definir parámetros funcionales que se transmitirán a los clientes. Estos parámetros se publican mediante la directiva **option**.

Declaración de opciones en el archivo dhcpd.conf

```
option domain-name sufijo;
option domain-name-servers servidores_dns;
option nis-domain dominio_nis;
option nis-servers servidores_nis;
```

dhcpd.conf: declaración de opciones	
sufijo	Sufijo DNS para los clientes.
servidores_dns	Servidor DNS utilizado por los clientes. Si hay varios, deberán separarse por comas.
dominio_nis	Cada vez más en desuso. Dominio NIS para los clientes.
servidores_nis	Cada vez más en desuso. Servidor NIS utilizado por los clientes. Si hay varios, deberán separarse por comas.

c. Declaración de los rangos de direcciones

Las direcciones que se asignarán se definen en una o varias secciones **subnet** del archivo**dhcpd.conf**. Dentro de estas secciones, aparte de los rangos de direcciones se definen las opciones DHCP que se enviarán junto a las ofertas de dirección. Las opciones más comunes son la puerta de enlace predeterminada y los servidores DNS que se utilizarán. Las opciones pueden declararse fuera o dentro de las secciones **subnet**, afectando al conjunto de direcciones asignadas o solamente a las del rango de subred en el que están respectivamente. Si se produce un conflicto (se ha declarado la misma opción en la configuración general y dentro de una sección subnet), es la opción de la subnet la que prevalece.

Declaración de una subnet en el archivo dhcpd.conf

```
subnet red netmask máscara {
  range inicio fin;
  option routers router;
}
```

dhcpd.conf: declaración de subnet	
red	La dirección de red en la que se encuentran las direcciones que se asignarán.
máscara	Máscara asociada a la red especificada.
inicio	Definición del rango de direcciones que se ofrecerá a los clientes. La primera dirección del rango.
fin	Definición del rango de direcciones que se ofrecerá a los clientes. La última dirección del rango.
router	La puerta de enlace predeterminada asociada a las direcciones propuestas.

d. Parámetros específicos a una máquina

Se puede asignar opciones particulares de manera específica a una máquina. Esta máquina será objeto de una declaración particular con la directiva **host**, parecida a una sección **subnet** pero dedicada a una sola máquina.

Se puede utilizar este método para asignar de forma específica a un host de la red una dirección IP fija para una máquina que, aunque disponga de un cliente DHCP, debería usar automáticamente la misma dirección. Por ejemplo, en el caso de una impresora de red cuya interfaz de configuración sea tan incómoda que se decida usar una configuración dinámica y para la que la reserva dhcp garantice la asignación de la dirección deseada.

Reserva de direcciones en dhcpd.conf

```
host máquina {
  hardware ethernet dirección_mac;
  fixed-address dirección_ip;
  option routers router;
  option domain-name sufijo;
  option domain-name-servers servidor_dns;
}
```

dhcpd.conf: configuración de host	

máquina	Declaración de parámetros para un host. Si la identificación se realiza mediante la dirección MAC, el nombre *máquina* carece de importancia.
dirección_mac	La dirección MAC del host que se configura.
dirección_ip	Dirección IP que se asignará al host.

e. Servidor con múltiples interfaces

Los servidores DHCP que posean múltiples interfaces deberán restringir sus comunicaciones a las tarjetas de red correspondientes. Por ejemplo, si un servidor tiene una interfaz configurada en 10.11.12.1 y otra en 192.168.200.1 y ofrece direcciones en la subred 192.168.200.0, es evidente que sólo deberá atender a las peticiones que reciba y ofertar direcciones en la interfaz correspondiente (192.168.200.1). La única dificultad reside en que este elemento de configuración no se encuentra en **/etc/dhcpd.conf**, sino que está en **/etc/defaults/dhcp3-server**.

f. Visualización de las concesiones dhcp

El servidor DHCP conserva la información de cada una de las concesiones asignadas en el archivo**dhcpd.leases** que se encuentra en el directorio **/var/lib/dhcp/**.

Se puede acceder a este archivo en modo lectura, pero nunca se debería modificar.

Ejemplo del archivo dhcpd.leases

Observe cómo aparecen las horas de renovación y de expiración de la concesión DHCP.

```
lease {
  interface "eth0";
  fixed-address 192.168.1.51;
  option subnet-mask 255.255.255.0;
  option routers 192.168.1.254;
  option dhcp-lease-time 864000;
  option dhcp-message-type 5;
  option domain-name-servers 194.2.0.20,194.2.0.50;
  option dhcp-server-identifier 192.168.1.1;
  renew 6 2010/07/10 14:55:34;
  rebind 3 2010/07/14 14:33:58;
  expire 4 2010/07/15 20:33:58;
}
```

3. Configuración del cliente

El comando **dhclient** permite realizar peticiones DHCP en los equipos cliente. Si el comando no se ejecuta manualmente por un administrador, se puede llamar mediante los scripts de inicialización de red. Si el equipo cliente no tiene dirección IP, realizará todos los pasos del procedimiento de petición DHCP. En caso contrario, solicitará al servidor una renovación de la concesión. También se puede usar **dhclient** para liberar una dirección asignada anteriormente por un servidor DHCP.

Ejemplo de uso de dhclient para solicitar una dirección

Observe las distintas etapas de asignación de la dirección: DHCPDISCOVER, DHCPOFFER, DHCPREQUEST y DHCPACK.

Ejemplo de

```
root@servidor# dhclient eth1
Internet Systems Consortium DHCP Client V3.1.3
Copyright 2004-2009 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/

Listening on LPF/eth1/00:22:68:98:8a:da
Sending on   LPF/eth1/00:22:68:98:8a:da
Sending on   Socket/fallback
DHCPDISCOVER on eth1 to 255.255.255.255 port 67 interval 8
DHCPOFFER of 172.18.142.243 from 172.18.142.225
DHCPREQUEST of 172.18.142.243 on eth1 to 255.255.255.255 port 67
```

```
DHCPACK of 172.18.142.243 from 172.18.142.225
bound to 172.18.142.243 -- renewal in 49 seconds.
root@servidor#
```

liberación de una dirección IP

Se puede comprobar que se envía un paquete DHCPRELEASE durante la ejecución del comando.

```
root@servidor# dhclient -r eth1
There is already a pid file /var/run/dhclient.pid with pid 2735
killed old client process, removed PID file
Internet Systems Consortium DHCP Client V3.1.3
Copyright 2004-2009 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/

Listening on LPF/eth1/00:22:68:98:8a:da
Sending on   LPF/eth1/00:22:68:98:8a:da
Sending on   Socket/fallback
DHCPRELEASE on eth1 to 172.18.142.225 port 67
root@servidor#
```

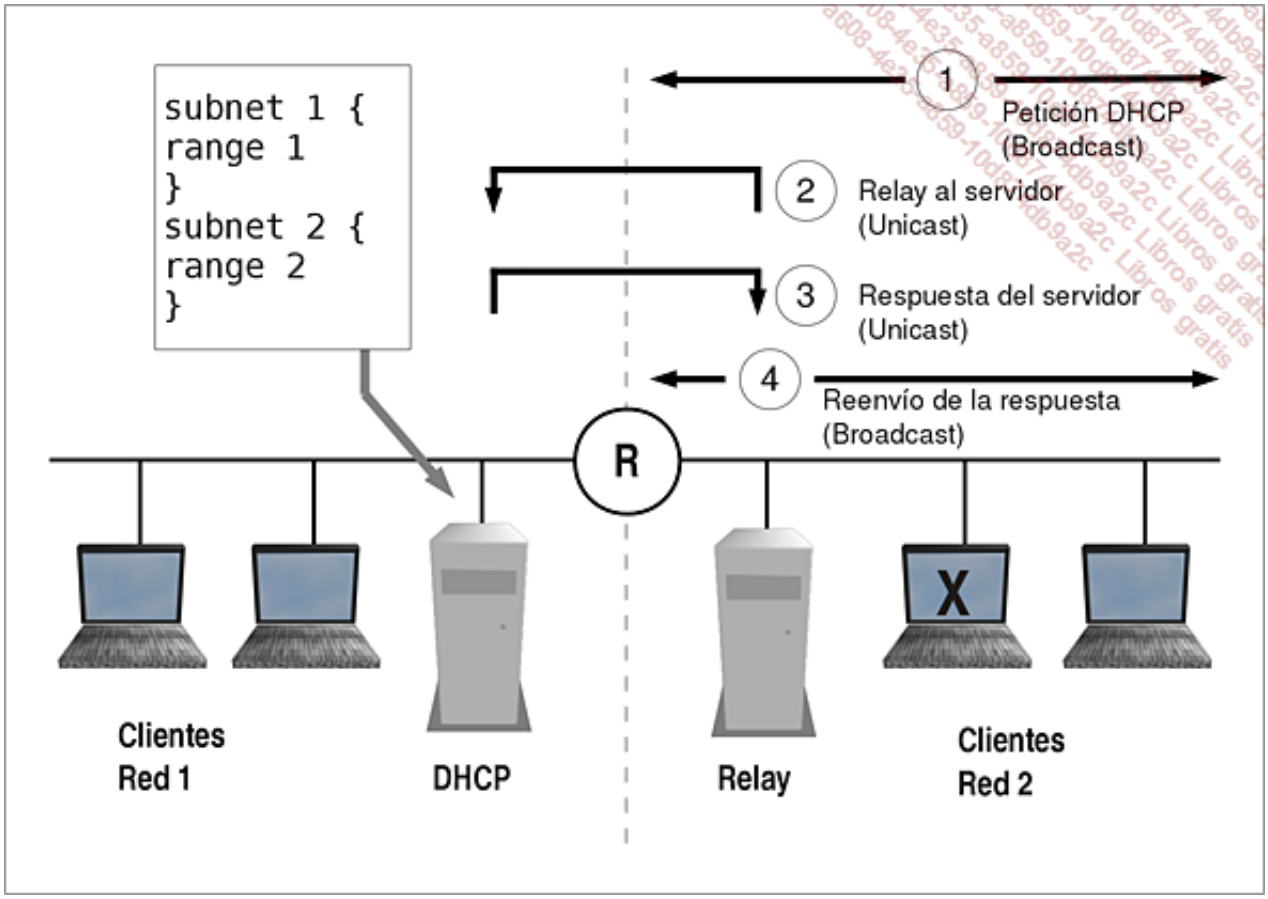
4. Agente de DHCP relay

Las comunicaciones DHCP se realizan por broadcast y los mensajes broadcast no pasan a través de los routers. Por consiguiente, tanto las peticiones DHCP como las respuestas de los servidores no producen ninguna acción fuera de la red local. La solución más fácil consiste evidentemente en poner un servidor DHCP en cada segmento de la red donde sean necesarios. Sin embargo, si se desea utilizar sólo un servidor para varias redes, existe una solución: los agentes DHCP relay.

a. Fundamentos del DHCP relay

La totalidad de la configuración DHCP, incluyendo la declaración de todos los elementos subnet y todos los rangos de direcciones, locales o remotos, se encontrará en un solo servidor DHCP. Una parte de los clientes, en cambio, se encontrará en un segmento de red diferente. Para que las comunicaciones puedan establecerse entre los clientes remotos y el servidor, el agente DHCP relay, que se encuentra también en el segmento, deberá procesar los broadcasts recibidos y retransmitir la petición en modo unicast al servidor DHCP. Los mensajes unicast pueden pasar por los routers, llegando la información a buen puerto. A continuación, el servidor DHCP responderá con un mensaje en modo unicast con el agente relay como destino y éste, a su vez, enviará un mensaje broadcast que recibirá el equipo cliente.

El cliente DHCP no sabe que está tratando con un agente relay, sino que piensa que hay un servidor DHCP real en su segmento.



b. Configuración de los agentes relay

El agente relay se inicia de forma interactiva mediante el comando **dhcrelay**. En la mayoría de distribuciones este comando se invoca desde un script de inicio de servicio y su configuración se obtiene a partir de un archivo de configuración.

Sintaxis del comando dhcrelay

```
dhcrelay -i interfaz dirección_servidor
```

dhcrelay: opciones y parámetros	
-i *interfaz*	Opcional. Especifica la interfaz por la que el agente relay estará a la escucha del servidor DHCP y de las peticiones de los clientes.
dirección_servidor	La dirección IP del servidor al que se transmitirán las peticiones DHCP.

Comprobación de los conocimientos adquiridos: preguntas/respuestas

Ponga a prueba sus conocimientos respondiendo a las preguntas siguientes. Estas preguntas no siempre esperan una respuesta cerrada. Las preguntas planteadas en la certificación, a pesar de que abordan los mismos temas, en su mayoría serán planteadas en forma de preguntas de opción múltiple o que requieran una respuesta corta, en palabras escritas en el teclado.

1. Preguntas

1. ¿Qué comando permite añadir una dirección IP secundaria a una interfaz?
2. Si una máquina tiene dos interfaces de red, ¿hay que configurar una segunda puerta de enlace predeterminada?
3. ¿Qué interés puede haber en informar el archivo /etc/ethers que asocia direcciones MAC con direcciones IP y cargarlo mediante el comando arp -f?
4. En el contexto de uso de los TCP Wrappers y del daemon tcpd, ¿cómo se resuelven los eventuales conflictos entre el archivo hosts.deny y el archivo hosts.allow?
5. ¿El comando ifconfig devuelve información acerca de una conexión WiFi?
6. ¿Por qué el parámetro -n se usa con frecuencia con los comandos arp, route o netstat?
7. Si un archivo abierto impide el desmontaje de un sistema de archivos, ¿cómo se puede encontrar el nombre del archivo en cuestión y el usuario que lo ha abierto?
8. ¿Para qué uso se ha impuesto la librería software libpcap?
9. La petición DHCP de un cliente se envía en forma de broadcast ya que el cliente no sabe la dirección del servidor DHCP. ¿Por qué la respuesta del servidor también se genera en broadcast?
10. Si se publica un servidor DNS en la configuración general de un servidor DHCP y hay otro servidor DNS publicado en una sección subnet, ¿qué servidor(es) DNS obtiene(n) los clientes de la subnet?

2. Respuestas

1. ¿Qué comando permite añadir una dirección IP secundaria a una interfaz?

   El comando ifconfig que, según los parámetros usados, puede asignar entre otros la dirección IP principal, la máscara de subred, la dirección MAC y una dirección IP secundaria si fuera necesario.

2. Si una máquina tiene dos interfaces de red, ¿hay que configurar una segunda puerta de enlace predeterminada?

   Por supuesto que no. Si se definen dos puertas de enlace predeterminadas en los archivos de configuración de las interfaces los scripts de inicialización de la red, que leen estos dos parámetros uno tras otro, sólo recordarán el último. De todas formas, como su nombre indica, la puerta de enlace predeterminada se utiliza en última instancia cuando el destino se encuentra en una red de la que no se conoce la ruta. Sin embargo, la tabla de enrutamiento es única e independiente de las interfaces: la puerta de enlace predeterminada, por tanto, debe ser un parámetro único e independiente de las interfaces.

3. ¿Qué interés puede haber en informar el archivo /etc/ethers que asocia direcciones MAC con direcciones IP y cargarlo mediante el comando arp -f?

   Si el archivo /etc/ethers se informa y se usa, el sistema aprende todas las asociaciones entre las direcciones MAC y las direcciones IP introducidas. Por consiguiente, toda comunicación con las mencionadas direcciones IP puede realizarse de forma directa, sin pasar previamente por una petición ARP. El beneficio neto es mínimo, las peticiones ARP son rápidas, pequeñas y en volumen insignificantes en comparación con el conjunto de las comunicaciones. Puede tener interés en términos de seguridad, ya que se evitan los broadcasts relacionados con estas peticiones haciendo que la red sea más discreta.

4. En el contexto de uso de los TCP Wrappers y del daemon tcpd, ¿cómo se resuelven los eventuales conflictos entre el archivo hosts.deny y el archivo hosts.allow?

   En principio, solamente uno de estos archivos debería existir. Si es hosts.allow, solamente los hosts mencionados en este archivo estarán autorizados, y si es hosts.deny, todos los hosts estarán autorizados salvo los que figuen en el archivo. En caso de conflicto, la solución más restrictiva es la que se aplica. Por lo

tanto, el archivo hosts.deny se ignorará.

5 ¿El comando ifconfig devuelve información acerca de una conexión WiFi?

En realidad, no. El comando ifconfig proporciona información sobre todas las interfaces del sistema, incluyendo las interfaces WiFi. Sin embargo, no proporciona ningún tipo de información relativa al funcionamiento inalámbrico: SSID, la calidad del señal o el tipo de cifrado. En cambio, el comando iwconfig es útil en este aspecto. Muestra la información relativa a una conexión WiFi e incluso permite configurarla.

6 ¿Por qué el parámetro -n se usa con frecuencia con los comandos arp, route o netstat?

Porque evita que el comando realice resoluciones de nombre inversas: estos comandos obtienen en principio datos numéricos sin tratar (direcciones IP, direcciones MAC y números de puerto). Sin embargo, por motivos estéticos, muestran por defecto los nombres correspondientes a estos datos numéricos, especialmente el nombre DNS que puede estar asociado a una dirección IP tratada. Pero si no existe ningún registro DNS para la dirección en cuestión, el comando intentará de todos modos realizar la resolución y no parará hasta que venza el timeout del resolver DNS. Por lo tanto, para un dudoso elemento de confort (no es seguro que el usuario prefiera los nombres a las direcciones), el comando pierde bastantes segundos intentando resolver las direcciones sin esperanzas. El uso de la opción -n evita que el comando realice estas tediosas búsquedas y, por tanto, disminuye el tiempo de respuesta.

7 Si un archivo abierto impide el desmontaje de un sistema de archivos, ¿cómo se puede encontrar el nombre del archivo en cuestión y el usuario que lo ha abierto?

Con el comando lsof, que devuelve los archivos abiertos, el nombre y el número del proceso responsable y el nombre del usuario propietario del proceso.

8 ¿Para qué uso se ha impuesto la librería software libpcap?

Para la captura de tramas. Son muchas las aplicaciones que la utilizan, entre las que se pueden citar especialmente tcpdump y wireshark. Una librería de código abierto permite a distintos programas utilizar el mismo formato de datos. Por tanto, se puede utilizar tcpdump para capturar la comunicación entre dos máquinas, wireshark para observarla detalladamente y un software de análisis para identificar las trazas características de un virus, por ejemplo.

9 La petición DHCP de un cliente se envía en forma de broadcast ya que el cliente no sabe la dirección del servidor DHCP. ¿Por qué la respuesta del servidor también se genera en broadcast?

Simplemente porque el cliente todavía no tiene una dirección IP y para enviar un mensaje unicast es necesaria una dirección de destino.

10 Si se publica un servidor DNS en la configuración general de un servidor DHCP y hay otro servidor DNS publicado en una sección subnet, ¿qué servidor(es) DNS obtiene(n) los clientes de la subnet?

El de la subnet. Los parámetros generales se usan para todos los clientes, excepto si hay otra información más concreta en una sección de la red.

Trabajos prácticos

1. Configuración de un servidor DHCP en el servidor alfa

Las estaciones de trabajo se multiplican en la red y la administración de direcciones IP se convierte en un problema. Decide instalar un servidor DHCP.

a. Configuración de una dirección IP fija para el servidor alfa

Comandos y archivos útiles

- /etc/network/interfaces
- /etc/resolv.conf
- ifdown
- ifup
- vi

Operaciones

1. Configurar el servidor alfa con una dirección IP fija. La dirección debe ser permanente y deberá conservarse después del reinicio. En los ejercicios se utilizará la dirección 192.168.200.101.
2. Verificar que el resolver use un servidor DNS válido.
3. Verificar que la dirección IP se use adecuadamente.
4. Verificar que la puerta de enlace por defecto se use adecuadamente.

Resumen de los comandos y resultado por pantalla

Archivo /etc/network/interfaces modificado:

Uso de la nueva configuración:

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug eth0
iface eth0 inet static
address 192.168.200.101
netmask 255.255.255.0
network 192.168.200.0
broadcast 192.168.200.255

gateway 192.168.200.254
```

Archivo /etc/resolv.conf:

```
alfa:~# ifdown eth0
alfa:~# ifdup eth0
alfa:~#
```

```
nameserver 194.2.0.20
nameserver 194.2.0.50
```

Comprobación de la dirección IP:

```
alfa:~# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 08:00:27:d2:22:f0
          inet addr:192.168.1.40  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fed2:22f0/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:5 errors:0 dropped:0 overruns:0 frame:0
          TX packets:36 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:300 (300.0 B)  TX bytes:3300 (3.2 KiB)
alfa:~#
```

Comprobación de la ruta por defecto usando dos comandos distintos:

b.

```
alfa:~# netstat -nr
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
192.168.200.0   0.0.0.0         25.5.255.255.0   U         0 0          0 eth0
0.0.0.0         192.168.200.254 0.0.0.0         UG        0 0          0 eth0
alpha:~# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.200.0   0.0.0.0         255.255.255.0   U     0      0        0 eth0
0.0.0.0         192.168.200.254 0.0.0.0         UG    0      0        0 eth0
alfa:~#
```

Instalación de los paquetes de software

En el servidor alfa, instale el servicio DHCP mediante el comando siguiente:

```
apt-get install isc-dhcp-server
```

Acepte las opciones por defecto. Si el reinicio del servicio da error, no se preocupe. La situación mejorará después de la configuración.

En la estación de trabajo, instale el software de captura de paquetes wireshark mediante el comando siguiente:

```
sudo apt-get install wireshark
```

c. Configuración del servicio

Directivas útiles

- option
- range
- subnet

Operaciones

1. En el archivo **/etc/dhcp/dhcpd.conf**, declare una red correspondiente a su dirección de red (192.168.200.0/24).
2. En el interior de la subnet, declare un rango de direcciones que comprenda de la**192.168.200.50** a la **192.168.200.99**.

3. En el interior de la subnet, declare **192.168.200.254** como dirección de la puerta de enlace por defecto.
4. En el interior de la subnet, declare su servidor DNS activo.
5. Configure la duración de las concesiones con un valor por defecto de 24h.
6. Reinicie el servicio (isc-dhcp-server)

Resumen de los comandos y resultado por pantalla

Archivo /etc/dhcp/dhcpd.conf modificado (esta sección debe añadirse al contenido ya existente del archivo):

```
default-lease-time 86400;
subnet 192.168.200.0 netmask 255.255.255.0 {
range 192.168.200.50 192.168.200.99;
option routers 192.168.200.254;
option domain-name-servers 192.168.200.254;
}
```

Reinicio del servicio:

```
alfa:/etc/dhcp# service isc-dhcp-server start
Starting ISC DHCP server:dhcpd.
alfa:/etc/dhcp/#
```

2. Uso del servicio DHCP

a. Configuración de la estación de trabajo

La estación de trabajo Ubuntu debe estar ya configurada como cliente DHCP. Para volver a solicitar de forma explícita una dirección IP basta con hacer clic en el icono que representa a la red en la barra superior de la pantalla. Un clic en **Auto eth0** provocará una solicitud de concesión DHCP.

Compruebe a continuación que la estación de trabajo ha obtenido correctamente una dirección y que esta dirección proviene del servidor alfa (podría ser necesario desactivar un posible servidor DHCP ya activo en la red).

Resumen de los comandos y resultado por pantalla

b.

```
usuario@estacion:~$ ifconfig eth0
eth0      Link encap:Ethernet  direcciónHW 00:22:25:d7:97:e6
          Direc. inet:192.168.200.50  Difus.:192.168.200.255  Másc:255.255.255.0
          Dirección inet6: fe80::222:15ff:fed7:97e6/64 Alcance:Enlace
          ACTIVO DIFUSIÓN FUNCIONANDO MULTICAST  MTU:1500  Métrica:1
          Paquetes RX:301626 errores:0 perdidos:0 overruns:0 frame:0
          Paquetes TX:186828 errores:0 perdidos:0 overruns:0 carrier:0
          colisiones:0 long.colaTX:1000
          Bytes RX:325369697 (325.3 MB) TX bytes:18332970 (18.3 MB)
          Memoria:d0180000-d01a0000

usuario@estacion:~$ cat /var/lib/dhcp/dhclient.leases
lease {
  interface "eth0";
  fixed-address 192.168.200.20;
  option subnet-mask 255.255.255.0;
  option routers 192.168.200.254;
  option dhcp-lease-time 864000;
  option dhcp-message-type 5;
  option domain-name-servers 212.27.40.241,212.27.40.240;
  option dhcp-server-identifier 192.168.200.254;
  renew 6 2010/07/10 14:55:34;
```

```
  rebind 3 2010/07/14 14:33:58;
  expire 4 2010/07/15 20:33:58;
}
usuario@estacion:~$
```

Reserva de una dirección IP para una impresora

Supongamos que tenemos una impresora cuya configuración de red es difícil de cambiar. Ante esta situación, el servidor DHCP nos brinda la solución. Sin embargo, por razones obvias de comodidad en la administración, esta impresora debe obtener sistemáticamente la misma dirección IP. Por lo tanto, decide reservarle una dirección IP.

Directivas útiles

- fixed-address
- hardware
- host
- option

Operaciones

1. En el archivo **/etc/dhcp/dhcpd.conf**, declare un host correspondiente a su impresora.
2. En la sección host, declare la dirección MAC de la impresora.
3. En la sección host, declare la dirección IP **192.168.200.11** para la impresora.
4. En la sección host, declare la puerta de enlace por defecto.

Extracto del archivo dhcpd.conf después de la configuración

c.

```
host printer1 {
hardware ethernet 00:12:34:56:78:9A;
fixed-ip-address 192.168.200.11;
option routers 192.168.200.254;
}
```

Captura de paquetes desde la estación de trabajo: comunicaciones DHCP

Su curiosidad natural le hace consultar con más detalle las comunicaciones DHCP entre el servidor y la estación de trabajo. Utilizará para ello las herramientas estándar de captura de paquetes tcpdump y wireshark.

Comandos útiles

- dhclient
- tcpdump
- wireshark

Operaciones

Los siguientes pasos se tienen que realizar en la estación de trabajo Ubuntu.

1. Desde un terminal, liberar la dirección IP obtenida previamente.
2. Desde otro terminal, capturar los paquetes con los puertos 67 y 68 como origen o destino (comunicaciones DHCP) por la interfaz eth0. Utilizar el comando tcpdump con privilegios de administrador.
3. Desde el primer terminal, provocar una petición DHCP.

4. Constatar el resultado por pantalla gracias a la salida estándar del comando tcpdump.
5. Repetir la operación, pero esta vez enviando el resultado al archivo **dhcp.cap**.
6. Abra el archivo con wireshark.
7. Observe el resultado de la captura.

Resumen de los comandos y resultado por pantalla

Terminal 1 - Liberación de la dirección IP:

```
usuario@estacion:~$ sudo dhclient -r eth0
There is already a pid file /var/run/dhclient.pid with pid 1998
killed old client process, removed PID file
Internet Systems Consortium DHCP Client V3.1.2
Copyright 2004-2008 Internet Systems Consortium.
All rights reserved.
For info, please visit http://www.isc.org/sw/dhcp/

Listening on LPF/eth0/08:00:27:7b:c8:79
Sending on   LPF/eth0/08:00:27:7b:c8:79
Sending on   Socket/fallback
DHCPRELEASE on eth0 to 192.168.200.254 port 67
usuario@estacion:~$ ifconfig eth0
eth0      Link encap:Ethernet  direcciónHW  08:00:27:7b:c8:79
          Dirección inet6: fe80::a00:27ff:fe7b:c879/64 Alcance:Enlace
          ACTIVO DIFUSIÓN FUNCIONANDO MULTICAST  MTU:1500  Métrica:1
          Paquetes RX:24856 errores:0 perdidos:0 overruns:0 frame:0
          Paquetes TX:6918 errores:0 perdidos:0 overruns:0 carrier:0
          colisiones:0 long.colaTX:1000
          Bytes RX:24613918 (24.6 MB) TX bytes:482889 (482.8 KB)
          Memoria:10 Dirección básica:0xd020
usuario@estacion:~$
```

Terminal 2 - Captura con tcpdump:

```
usuario@estacion:~$ sudo tcpdump -i eth0 -n port 67 and port 68
tcpdump: WARNING: eth0: no IPv4 address assigned
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
usuario@estacion:~$
```

Terminal 1 - Petición DHCP:

```
usuario@estacion:~$ sudo dhclient eth0
Internet Systems Consortium DHCP Client V3.1.2
Copyright 2004-2008 Internet Systems Consortium.
All rights reserved.
For info, please visit http://www.isc.org/sw/dhcp/

Listening on LPF/eth0/08:00:27:7b:c8:79
Sending on   LPF/eth0/08:00:27:7b:c8:79
Sending on   Socket/fallback
DHCPDISCOVER on eth0 to 255.255.255.255 port 67 interval 5
DHCPOFFER of 192.168.200.102 from 192.168.200.254
DHCPREQUEST of 192.168.200.102 on eth0 to 255.255.255.255 port 67
DHCPACK of 192.168.200.102 from 192.168.200.254
bound to 192.168.200.102 -- renewal in 329015 seconds.
usuario@estacion:~$
```

Terminal 2 - Resultado de tcpdump:

```
(...)
12:06:59.003789 IP 0.0.0.0.68 > 255.255.255.255.67: BOOTP/DHCP,
Request from 08:00:27:7b:c8:79, length 300
12:06:59.008562 IP 192.168.200.254.67 > 192.168.200.102.68: BOOTP/DHCP,
Reply, length 548
12:06:59.051798 IP 0.0.0.0.68 > 255.255.255.255.67: BOOTP/DHCP,
Request from 08:00:27:7b:c8:79, length 300
12:06:59.056980 IP 192.168.200.254.67 > 192.168.200.102.68: BOOTP/DHCP,
Reply, length 548
12:06:59.842693 IP 192.168.200.101.67 > 192.168.200.50.68: BOOTP/DHCP,
Reply, length 300
[ Ctrl - C ]
5 packets captured
6 packets received by filter
0 packets dropped by kernel
usuario@estacion:~$
```

Terminal 1 - Liberación de la dirección IP:

```
usuario@estacion:~$ sudo dhclient -r eth0
(...)
usuario@estacion:~$
```

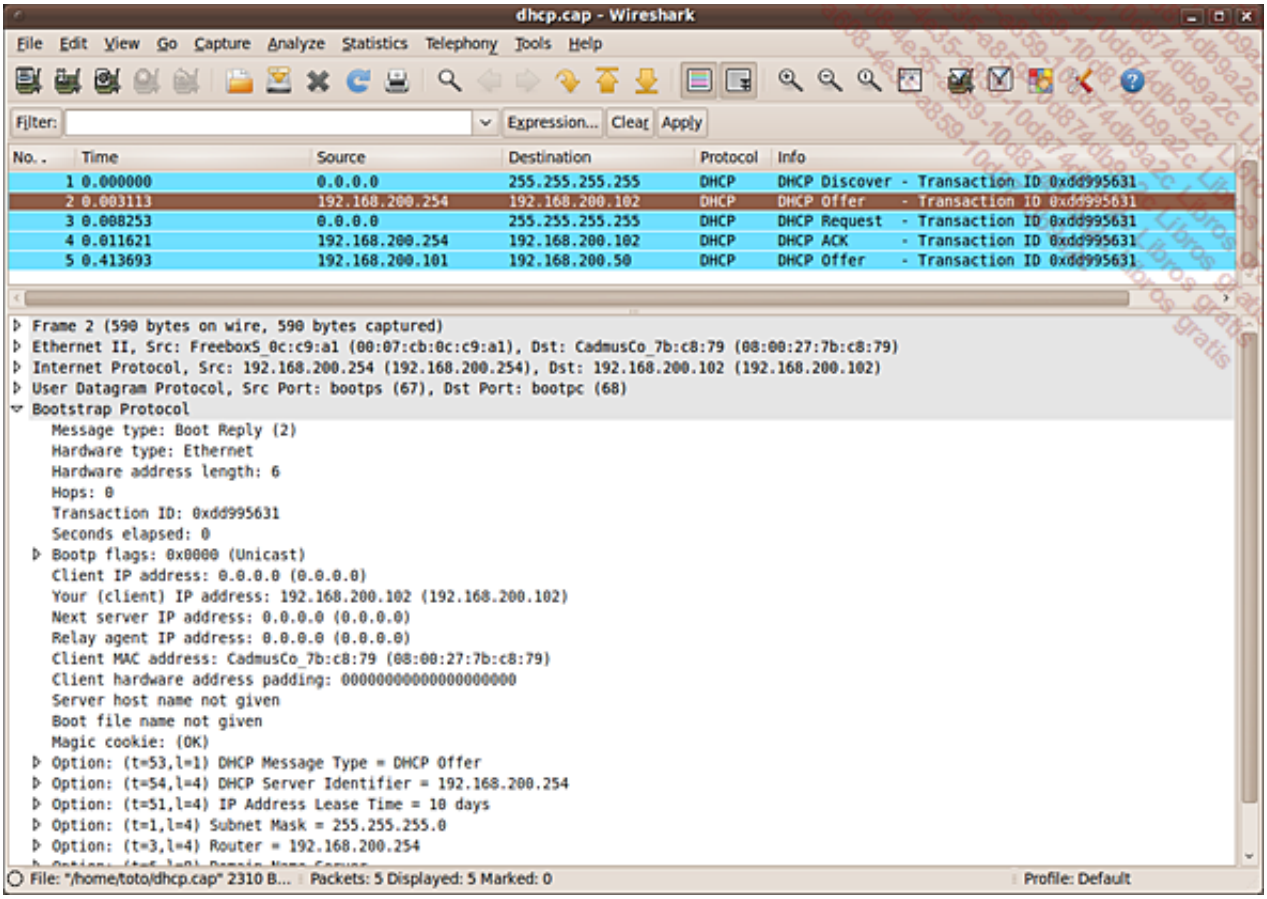
Terminal 2 - Captura con tcpdump y resultado en un archivo:

```
usuario@estacion:~$ sudo tcpdump -w dhcp.cap -i eth0 -n port 67 or port 68
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
[ Ctrl - C ]
usuario@estacion:~$
```

Terminal 1 - Petición DHCP:

```
usuario@estacion:~$ sudo dhclient eth0
(...)
usuario@estacion:~$
```

Observación del resultado con wireshark.


dhcp.cap - Wireshark
File Edit View Go Capture Analyze Statistics Telephony Tools Help
Filter: Expression... Clear Apply
No. . Time Source Destination Protocol Info
1 0.000000 0.0.0.0 255.255.255.255 DHCP DHCP Discover - Transaction ID 0xdd995631
2 0.003113 192.168.200.254 192.168.200.102 DHCP DHCP Offer - Transaction ID 0xdd995631
3 0.008253 0.0.0.0 255.255.255.255 DHCP DHCP Request - Transaction ID 0xdd995631
4 0.011621 192.168.200.254 192.168.200.102 DHCP DHCP ACK - Transaction ID 0xdd995631
5 0.413693 192.168.200.101 192.168.200.50 DHCP DHCP Offer - Transaction ID 0xdd995631
Frame 2 (590 bytes on wire, 590 bytes captured)
Ethernet II, Src: FreeboxS_0c:c9:a1 (00:07:cb:0c:c9:a1), Dst: CadmusCo_7b:c8:79 (08:00:27:7b:c8:79)
Internet Protocol, Src: 192.168.200.254 (192.168.200.254), Dst: 192.168.200.102 (192.168.200.102)
User Datagram Protocol, Src Port: bootps (67), Dst Port: bootpc (68)
Bootstrap Protocol
Message type: Boot Reply (2)
Hardware type: Ethernet
Hardware address length: 6
Hops: 0
Transaction ID: 0xdd995631
Seconds elapsed: 0
Bootp flags: 0x0000 (Unicast)
Client IP address: 0.0.0.0 (0.0.0.0)
Your (client) IP address: 192.168.200.102 (192.168.200.102)
Next server IP address: 0.0.0.0 (0.0.0.0)
Relay agent IP address: 0.0.0.0 (0.0.0.0)
Client MAC address: CadmusCo_7b:c8:79 (08:00:27:7b:c8:79)
Client hardware address padding: 00000000000000000000
Server host name not given
Boot file name not given
Magic cookie: (OK)
Option: (t=53,l=1) DHCP Message Type = DHCP Offer
Option: (t=54,l=4) DHCP Server Identifier = 192.168.200.254
Option: (t=51,l=4) IP Address Lease Time = 10 days
Option: (t=1,l=4) Subnet Mask = 255.255.255.0
Option: (t=3,l=4) Router = 192.168.200.254
File: "/home/toto/dhcp.cap" 2310 B... Packets: 5 Displayed: 5 Marked: 0 Profile: Default

Requisitos y objetivos

1. Requisitos

Los conocimientos adquiridos con la certificación LPI nivel 1, especialmente:

- ☒ Conocer la estructura del archivo /etc/passwd.
- ☒ Conocer la existencia y los fundamentos del archivo hosts.

2. Objetivos

Al final de este capítulo, usted será capaz de:

- ☒ Interpretar una configuración NSS.
- ☒ Comprender la autentificación modular PAM.
- ☒ Conocer los principales módulos PAM.
- ☒ Modificar la configuración PAM para permitir cambios en la forma de autentificarse.
- ☒ Conocer el formato de los archivos LDIF.
- ☒ Consultar un directorio LDAP.
- ☒ Administrar las contraseñas en un directorio OpenLDAP.
- ☒ Añadir o modificar elementos de un directorio OpenLDAP.
- ☒ Configurar la autentificación de un sistema Linux en un directorio OpenLDAP.

Evolución de la autentificación

1. Los primeros sistemas Unix y el archivo passwd

a. Contraseñas en el archivo /etc/passwd

Desde su aparición, los sistemas Unix utilizan el archivo **/etc/passwd** como base de datos de cuentas de usuarios. Este archivo se utiliza de forma natural para abrir sesiones en el sistema. Como su nombre todavía indica albergaba, además de los identificadores de los usuarios, sus contraseñas encriptadas. Si algún otro elemento distinto del de apertura de sesión necesita información sobre las cuentas (conexión ftp, apertura de sesión remota, etc.), también consultará este archivo. En esta sencilla situación inicial, hay una única base de datos de cuentas de usuario y múltiples aplicaciones que la usan. Todas las aplicaciones tienen que reconocer el formato de esta base de datos.

b. Contraseñas en el archivo /etc/shadow

Con la evolución de las técnicas de ataque de contraseñas, se hizo necesario mover las contraseñas a un archivo no accesible a usuarios normales. Para ello, se almacenaron en un nuevoarchivo: **/etc/shadow** cerrado a los usuarios. Los parámetros de autentificación con shadow se administran mediante el archivo **/etc/login.defs**. Los parámetros almacenados en este archivo son en general adecuados.

Gestión de errores de autentificación en el archivo login.defs

De la gran cantidad de parámetros del archivo login.defs, los que están relacionados con el login son los que se modifican con mayor frecuencia.

```
usuario@ubuntu:~$ grep LOGIN /etc/login.defs
LOGIN_RETRIES 5
LOGIN_TIMEOUT 60
usuario@ubuntu:~$
```

2. Otras bases de datos

En la consulta de datos de identificación, la situación se complicó cuando aparecieron otras bases de datos de cuentas diferentes del archivo **passwd** y, sobre todo, más complejas. A menudo, estas bases de datos de identidad están centralizadas, como es el caso de NIS (*Network Information Server*) o LDAP (*Lightweight Directory Access Protocol*). La primera solución propuesta fue, por supuesto, volver a escribir los programas que originalmente operaban con el archivo **/etc/passwd**para que fueran capaces de consultar las bases de datos centralizadas en red. Este método carecía claramente de flexibilidad debido a que obligaría a rehacer una gran cantidad de programas en profundidad cada vez que apareciese un cambio o una nueva forma de almacenamiento en las bases centralizadas.

3. NSS

NSS (*Name Service Switch*) es una primera respuesta a la multiplicidad de bases de datos locales o centralizadas. NSS tiene como objetivo normalizar la resolución de nombres en un sistema. NSS permite resolver un nombre obteniendo la información asociada, como por ejemplo un nombre de usuario y su uid, un nombre de grupo y su gid o incluso un nombre de host y su dirección IP.

En el funcionamiento de NSS, el archivo **/etc/nsswitch.conf** determina para distintos tipos de resoluciones la fuente de información que se debe consultar. Las aplicaciones que necesiten esta información consultarán las fuentes en el orden impuesto por el archivo **nsswitch.conf**. De este modo, la resolución se apoya en librerías NSS (**libnss_*X*.so** donde *X* representa el servicio de resolución empleado) y las aplicaciones no necesitan conocer directamente el método de resolución empleado.

Formato del archivo nsswitch.conf

```
resolución: fuente_1 fuente_n
```

nsswitch.conf: formato del archivo	
resolución	El tipo de resolución que se realizará.
fuente_1	Obligatorio. La primera fuente de resolución que se usará.
fuente_n	Opcional. La o las otras fuentes de resolución posibles que se utilizarán después de la primera.

Ejemplo de archivo

nsswitch.conf

En este ejemplo se puede ver que las resoluciones de tipo password, group y shadow se realizarán mediante la librería libnss_compat.so y que la resolución de nombres de host se realizará mediante las librerías libnss_files.so y libnss_dns.so. Esto significa que los elementos de identificación de los usuarios se encontrarán en los archivos locales de /etc y que la resolución de nombres de host se realizará en primer lugar mediante el archivo local (/etc/hosts) antes de utilizar el servicio dns.

```
passwd:         compat
group:          compat
shadow:         compat

hosts:          files dns
networks:       files

protocols:      db files
services:       db files
ethers:         db files
rpc:            db files

netgroup:       nis
```

En un sistema Linux moderno, NSS ya sólo se usa para operaciones de identificación, es decir, encontrar información de una entidad. Todo lo relativo a la autentificación se realiza en un mecanismo más elaborado: PAM.

4. Módulos de autentificación

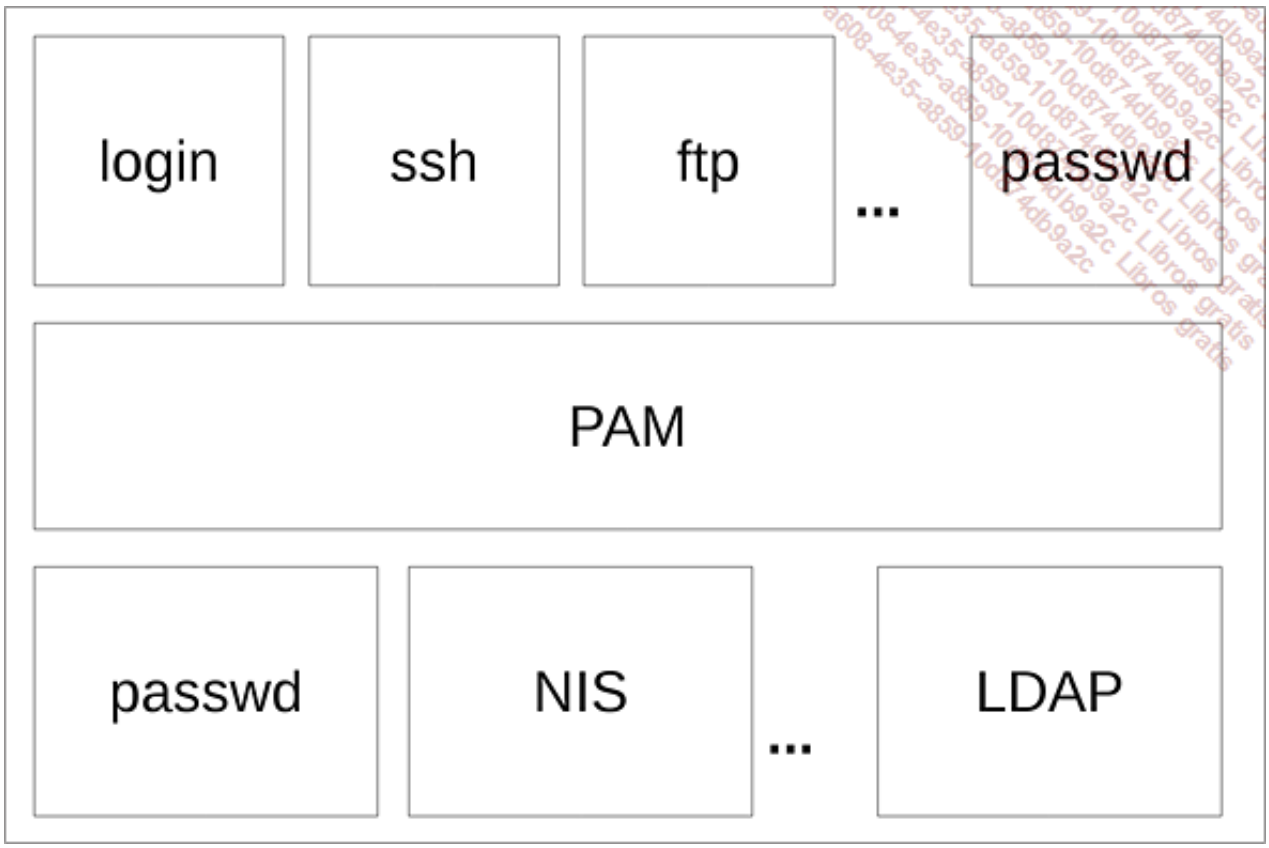
Si NSS ya representa un progreso en relación a los archivos estáticos usados en los primeros años, la revolución llega con PAM (*Pluggable Authentication Module*). PAM es un mecanismo complementario de NSS que proporciona una autentificación a medida mediante la ejecución de módulos a la elección del administrador.

Cuando se abre una sesión en Linux, el usuario tiene que presentar su identificador y una contraseña. Gracias a la resolución NSS, se deducirán los identificadores uid/gid, así como el resto de parámetros necesarios (fecha de expiración, etc.). En lo que respecta a PAM, ejecutará según su configuración un módulo u otro para proporcionar la autentificación, pero también puede realizar ciertas tareas ligadas a la apertura de sesión, como por ejemplo la definición de variables.

PAM

1. El principio

PAM se posiciona como un intermediario entre las aplicaciones y los métodos de autentificación.



El objetivo principal de PAM es proporcionar una capa de abstracción entre las aplicaciones y los métodos de autentificación. De este modo, una aplicación que quiera ser flexible y evolutiva en cuanto a los métodos de autentificación que emplea sólo deberá ser compatible con PAM. Esto significa que deberá ser capaz de dirigirse a la capa de autentificación PAM, sin importarle todo lo que hay detrás. Paralelamente, los procedimientos de autentificación, sean cuales sean, deberán ser accesibles y utilizables por el mecanismo PAM.

Una aplicación solicita a PAM si un usuario se puede conectar. PAM, en función de su configuración, invocará los módulos que utilizarán un método de autentificación. Si el resultado es positivo (el usuario ha proporcionado los elementos correctos de autentificación), PAM devuelve la autorización de conexión a la aplicación.

PAM tiene otra ventaja. Acabamos de ver que la solicitud de autentificación entrañaba la carga de módulos. Pues bien, el número de módulos no tiene límites y éstos se pueden acumular. Por lo tanto, se puede solicitar una doble autentificación siguiendo dos métodos de autentificación distintos. Además, se puede sacar provecho de la autentificación con PAM para provocar la carga de librerías sin relación con la autentificación. Por lo tanto, es posible desencadenar muchas acciones una vez que se ha realizado con éxito la autentificación.

En resumen: cuando se solicita a un usuario que se autentifique, los módulos PAM se cargan en función de un archivo de configuración y estos módulos provocan ciertas acciones, que pueden ser la propia autentificación u otras acciones.

2. Los módulos PAM

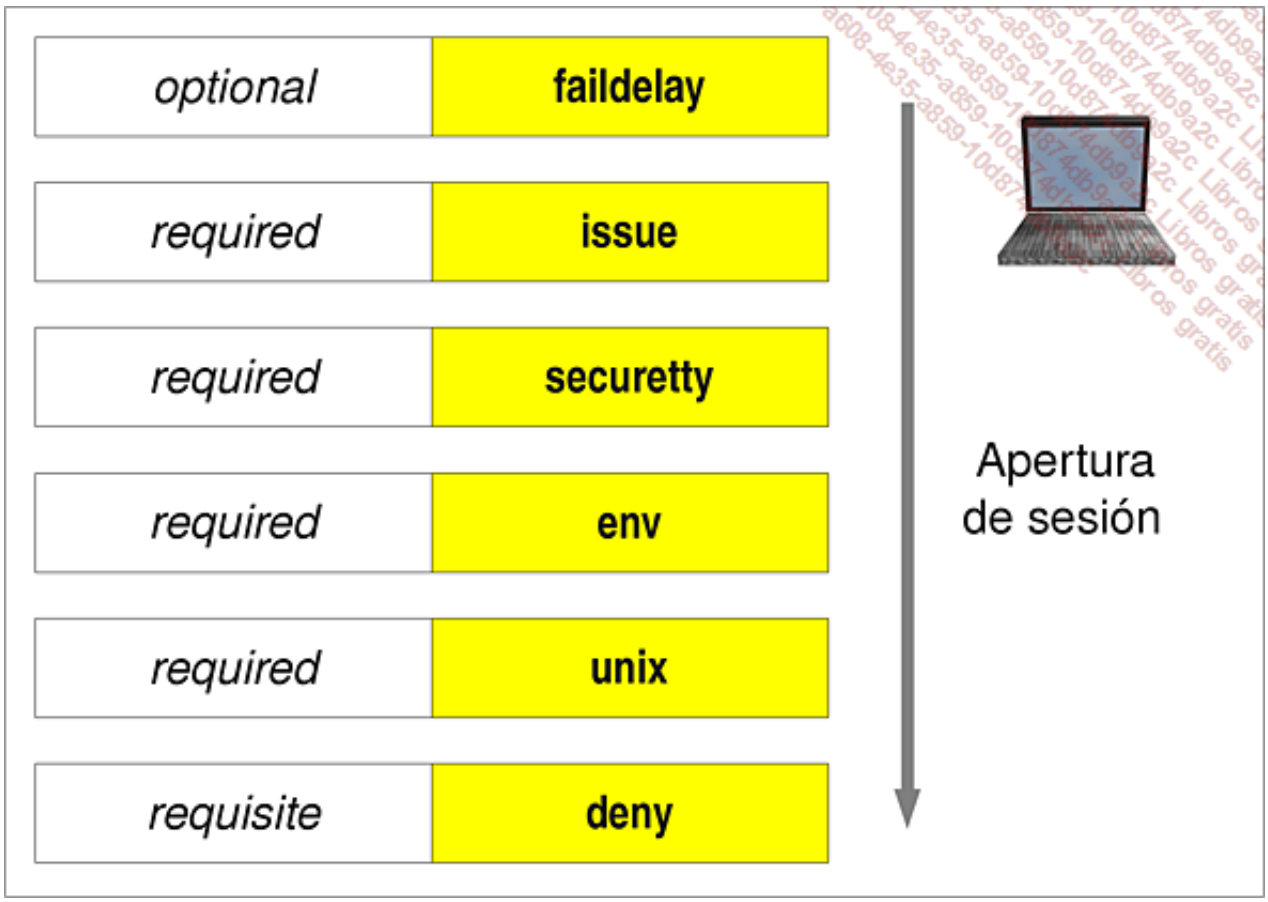
a. Los módulos PAM principales

Los módulos PAM, invocados cuando se producen operaciones de autentificación, son muchos y están enfocados a distintos usos. Algunos de ellos, sin embargo, se utilizan con mucha frecuencia y hay que conocerlos. Otros son menos frecuentes y dependen de la distribución que se esté usando. No obstante, conocer su funcionamiento y su finalidad permite comprender mejor la mecánica y la filosofía de PAM.

Estos módulos están en archivos cuya ubicación estándar es **/lib/security**.

Módulos PAM principales	
pam_securetty.so	Prohíbe el login para la cuenta root excepto en los terminales listados en /etc/securetty.
pam_nologin.so	Si el archivo /etc/nologin existe, muestra su contenido ante cualquier intento de apertura de sesión y prohíbe el login ante cualquier usuario que no sea root.
pam_env.so	Declara las variables de entorno que se leen en /etc/environment o en el archivo al que se hace referencia con el parámetro "envfile=".
pam_unix.so	Permite la autentificación mediante el método tradicional de los archivos /etc/passwd y /etc/shadow.
pam_deny.so	Vía muerta. Generalmente se ejecuta si ningún otro módulo se ha ejecutado con éxito.
pam_permit.so	Devuelve un resultado positivo incondicionalmente.
pam_limits.so	Asigna ciertas limitaciones funcionales a usuarios o grupos en función de los datos del archivo /etc/security/limits.conf.
pam_cracklib.so	Se asegura que la contraseña empleada presenta un nivel de seguridad suficiente.
pam_selinux.so	Si selinux está activo en el sistema, este módulo va a asegurar que el shell se ejecuta en el contexto de seguridad adecuado.
pam_lastlog.so	Muestra la información de la última apertura de sesión con éxito.
pam_mail.so	Comprueba la presencia de nuevos correos para un usuario (mensajería interna).

b. Funcionamiento en pilas de módulos



Para una acción determinada, por ejemplo la autentificación, se invocan varios módulos PAM. Se habla entonces de una pila de módulos PAM. El funcionamiento en pila es una de las mayores aportaciones de los

servicios PAM.

3. Configuración de PAM

a. Estructura de los archivos de configuración

Las primeras versiones de PAM tenían su configuración en el archivo **/etc/pam.conf**. La gran complejidad de PAM ha hecho necesaria de forma inminente una estructura más modular en sus elementos de configuración. Casi la totalidad de las implementaciones actuales utilizan un directorio **/etc/pam.d** que contiene tantos archivos como aplicaciones que usan PAM. Si existe eldirectorio **/etc/pam.d**, el archivo **/etc/pam.conf** no se consultará.

Cada aplicación que utilice PAM necesita un archivo (en general del mismo nombre que la aplicación) que alberga su configuración PAM.

Formato de un archivo de /etc/pam.d

El archivo contendrá tantas líneas como módulos se deseen llamar. Todas las líneas seguirán la siguiente estructura:

```
tipo control módulo argumentos
```

Archivo de pam.d: formato estándar	
tipo	Representa el tipo de acción que necesita recurrir a PAM. Los cuatro valores posibles son: **auth**, **account**, **password** y **session**.
control	Indica cómo deberá reaccionar el módulo ante el éxito o el error de su ejecución. Los valores comunes son **required**, **requisite**, **sufficient** y**optional**.
módulo	El nombre del módulo invocado. El formato estándar es:**pam_*servicio.so***, donde *servicio* representa el nombre actual del módulo.
argumentos	Parámetros opcionales enviados al módulo para modificar su funcionamiento.

Los posibles valores de tipo y de control se explicarán con más detalle, pero con lo explicado hasta ahora ya se puede entender la estructura del archivo de configuración. En el extracto mostrado a continuación se puede ver que la línea sólo afecta a las operaciones de autentificación (auth), que la ejecución del módulo es obligatoria (required), que el módulo usa el método de autentificación tradicional unix, es decir, los archivos passwd y shadow (pam_unix.so) y, finalmente, que este módulo debe aceptar una autentificación realizada con una contraseña vacía (nullok). Observe que el parámetro nullok es específico del módulo y que cada módulo soportará todos los parámetros que su desarrollador haya querido.

Extracto de un archivo de configuración pam para la aplicación login

En este ejemplo se trata la autentificación (auth), la ejecución del módulo es obligatoria (required) y el módulo usa el archivo de contraseñas clásico (pam_unix.so). Por último, se autoriza el uso de una contraseña vacía, tal y como indica el argumento (nullok).

```
auth required pam_unix.so nullok
```

b. Tipos de acción de PAM

Cada línea de un archivo de configuración PAM debe comenzar por una de estas cuatro palabras clave que determinan en qué tipo de acción debe actuar el módulo.

- auth: la acción de autentificación propiamente dicha. Los módulos llamados con la acción auth se ejecutan para o durante la autentificación.
- account: acceso a la información de cuentas no relacionada con la autentificación.

- session: acciones que se deben realizar antes o después de la apertura de la sesión.
- password: gestión de contraseñas.

Extracto del archivo de configuración pam para la aplicación login

Este ejemplo es un extracto muy resumido de un archivo de login estándar que ilustra tanto el concepto de pila como la naturaleza modular de PAM.

Al principio, aparecen dos módulos invocados para la autentificación: pam_securetty aprovecha la autentificación para verificar que la cuenta no es la del superusuario y pam_unix es quien realmente realiza la autentificación a partir del archivo /etc/passwd.

El mismo módulo pam_unix también se declara con el tipo account. Si las aplicaciones que son compatibles com PAM necesitan información sobre las cuentas de los usuarios, necesitarán el módulo pam_unix usado con el tipo account.

El módulo pam_env se invoca con el tipo session, lo que asegura su ejecución (y por tanto la declaración de variables) en la sesión del usuario.

El módulo pam_cracklib se invoca con el tipo password. Si una aplicación de gestión de contraseñas compatible con PAM desea modificar una contraseña debe pasar el control de complejidad realizado por el módulo cracklib.

```
auth      required    pam_securetty.so
auth      required    pam_unix.so
account   required    pam_unix.so
session   required    pam_env.so readenv=1 envfile=/etc/default/locale
password  required    pam_cracklib.so retry=3 minlen=6
```

c. Tipos de control de los módulos

Los módulos se invocan con un "control_flag" (indicador de control) que determina el comportamiento en caso de error o éxito del módulo.

Este elemento obligatorio es el segundo campo de la línea de configuración.

- required: el módulo debe devolver de forma obligatoria un valor de éxito. Si un módulo de autentificación es required, su error impide la apertura de sesión. Sin embargo, el resto de módulos de la pila se ejecutarán.
- requisite: el módulo debe devolver de forma obligatoria un valor de éxito. Si un módulo de autentificación es requisite, su error impide la apertura de sesión. Además, el resto de módulos de la pila no se ejecutarán.
- sufficient: si el módulo se ejecuta con éxito y ningún módulo required o requisite ha dado error, el resto de módulos de la pila se ignorarán.
- optional: el módulo puede ejecutarse con éxito o error sin influenciar al resto de la pila. Es decir, si un módulo opcional da error y un módulo required de la misma pila se ejecuta con éxito, entonces el resultado global de la ejecución de la pila es positivo.

Ejemplos de archivos de configuración PAM

A continuación se muestran dos archivos PAM. El primero (gdm) gestiona la apertura de sesión gráfica en el entorno Gnome y el otro (gdm-autologin) asegura la apertura automática sin contraseña de la sesión gráfica. Para ver las diferencias entre estos dos modos de funcionamiento en lo que respecta a la autentificación de usuarios en este ejemplo sólo tienen interés los módulos declarados en las líneas de tipo auth.

Los primeros módulos que se cargan, pam_nologin y pam_env, son comunes en ambos archivos. A título informativo, pam_nologin prohíbe la conexión de los usuarios normales si el archivo /etc/nologin existe y ha sido rellenado por el administrador y pam_env define diversas variables en el momento de la autentificación.

A continuación, el archivo gdm incluye el subarchivo common-auth que invocará los elementos de autentificación deseados en el sistema (como mínimo pam_unix para la autentificación tradicional) y después carga el módulo pam_gnome_keyring que permitirá a los usuarios debidamente autentificados en Gnome acceder a ciertas características que necesitarían, por lo general, volver a autentificar al usuario.

El archivo gdm-autologin, en cambio, sólo carga un módulo: pam_permit que siempre devuelve un

resultado positivo, cuya ejecución es obligatoria (el módulo es required) y por lo tanto autorizará la apertura de sesión incondicionalmente.

El archivo de configuración pam para la apertura de sesión manual Gnome: gdm

```
auth    requisite       pam_nologin.so
auth    required        pam_env.so readenv=1
auth    required        pam_env.so readenv=1 envfile=/etc/default/locale
@include common-auth
auth    optional        pam_gnome_keyring.so
@include common-account
session [success=ok ignore=ignore module_unknown=ignore default=bad] pam_selinux.so close
session required        pam_limits.so
@include common-session
session [success=ok ignore=ignore module_unknown=ignore default=bad] pam_selinux.so open
session optional        pam_gnome_keyring.so auto_start
@include common-password
```

El archivo de configuración pam para la apertura de sesión automática Gnome: gdm-autologin

```
auth    requisite       pam_nologin.so
auth    required        pam_env.so readenv=1
auth    required        pam_env.so readenv=1 envfile=/etc/default/locale
auth    required        pam_permit.so
@include common-account
session [success=ok ignore=ignore module_unknown=ignore default=bad] pam_selinux.so close
session required        pam_limits.so
@include common-session
session [success=ok ignore=ignore module_unknown=ignore default=bad] pam_selinux.so open
@include common-password
```

LDAP

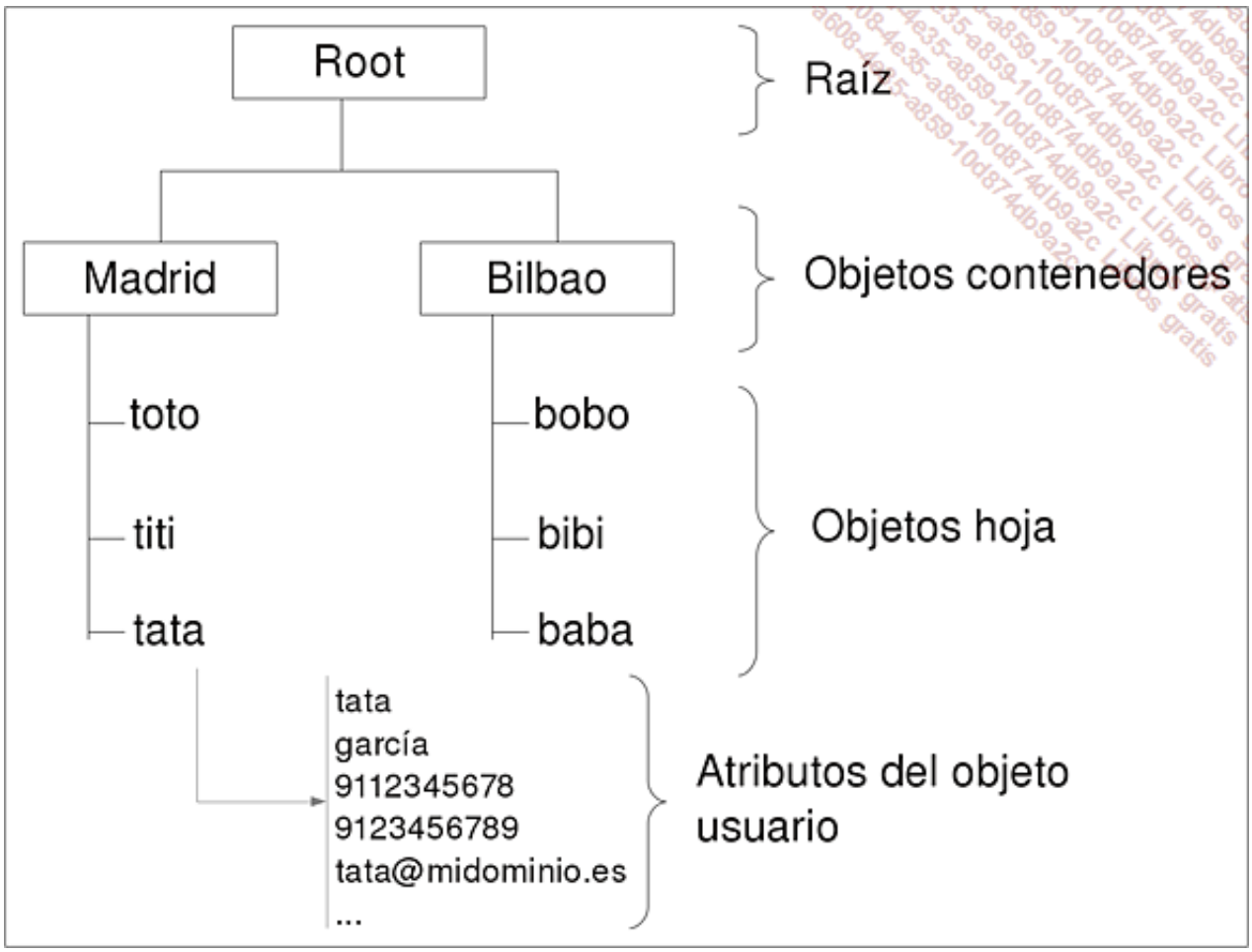
1. Características generales

a. Los directorios

En 1990, la ITU (*International Telecommunication Union*) propuso una norma de estructuración de los directorios electrónicos. Esta norma que tiene como objetivo establecer un marco de funcionamiento y de referencia común para todos los desarrolladores se llama X500.

Los primeros programas que usaron esta norma fueron evidentemente los mensajes electrónicos. El NDS (*Netware Directory Services*), célebre en su época, fue el primer uso relevante de las tecnologías de directorios X500 al servicio de un sistema operativo en red. Hoy en día los directorios están ampliamente extendidos, ya sean internos a un sistema operativo en red (como es el Active Directory de Microsoft) o bien estén a disposición de otras aplicaciones. En ese caso, se suele hablar de directorios de "páginas blancas".

b. Estructura y terminología



Los directorios electrónicos X500 presentan características estructurales comunes. Son jerárquicos y tienen un punto de origen que generalmente se denomina Root. Todo elemento del directorio se llama objeto; algunos elementos son estructurales y otros son totalmente informativos. Los elementos estructurales se llaman contenedores y son de distintos tipos, como por ejemplo organización, dominio o unidad organizativa.

Todo objeto del directorio alberga en su interior información en distintos formatos. Estos datos se denominan atributos del objeto.

c. Esquema

Los directorios fueron originalmente diseñados para almacenar y administrar identidades. Naturalmente, en ellos se encuentran los objetos que representan a las personas y los atributos para identificarlas y definirlas tales como el nombre, los apellidos, el teléfono y la dirección electrónica. El conjunto de tipos de objetos

posibles en el directorio y para cada objeto el conjunto de atributos se definen en el esquema del directorio.

Sin embargo, es natural que un desarrollador o un usuario quieran almacenar en su directorio información de índole particular para cubrir las necesidades propias de sus aplicaciones. Si el esquema original no lo permite, entonces se puede realizar una extensión del esquema. La extensión del esquema consiste en definir para un directorio nuevos tipos de objetos o nuevos atributos para un tipo de objeto existente. Por ejemplo, si una empresa tiene un directorio con el censo de su personal y dicho personal debe llevar calzado de protección, puede ser más interesante extender el esquema para añadir a los objetos de tipo persona el atributo "pie" que gestionar una lista más o menos actualizada en una hoja de cálculo.

El tipo de cada objeto (unidad organizativa, usuario, grupo, etc.) se llama clase. Una clase de objetos se define por el conjunto de atributos que la componen. Entre todos estos atributos, hay uno que tendrá una importancia particular en la denominación del objeto, es el CN (*Common Name*).

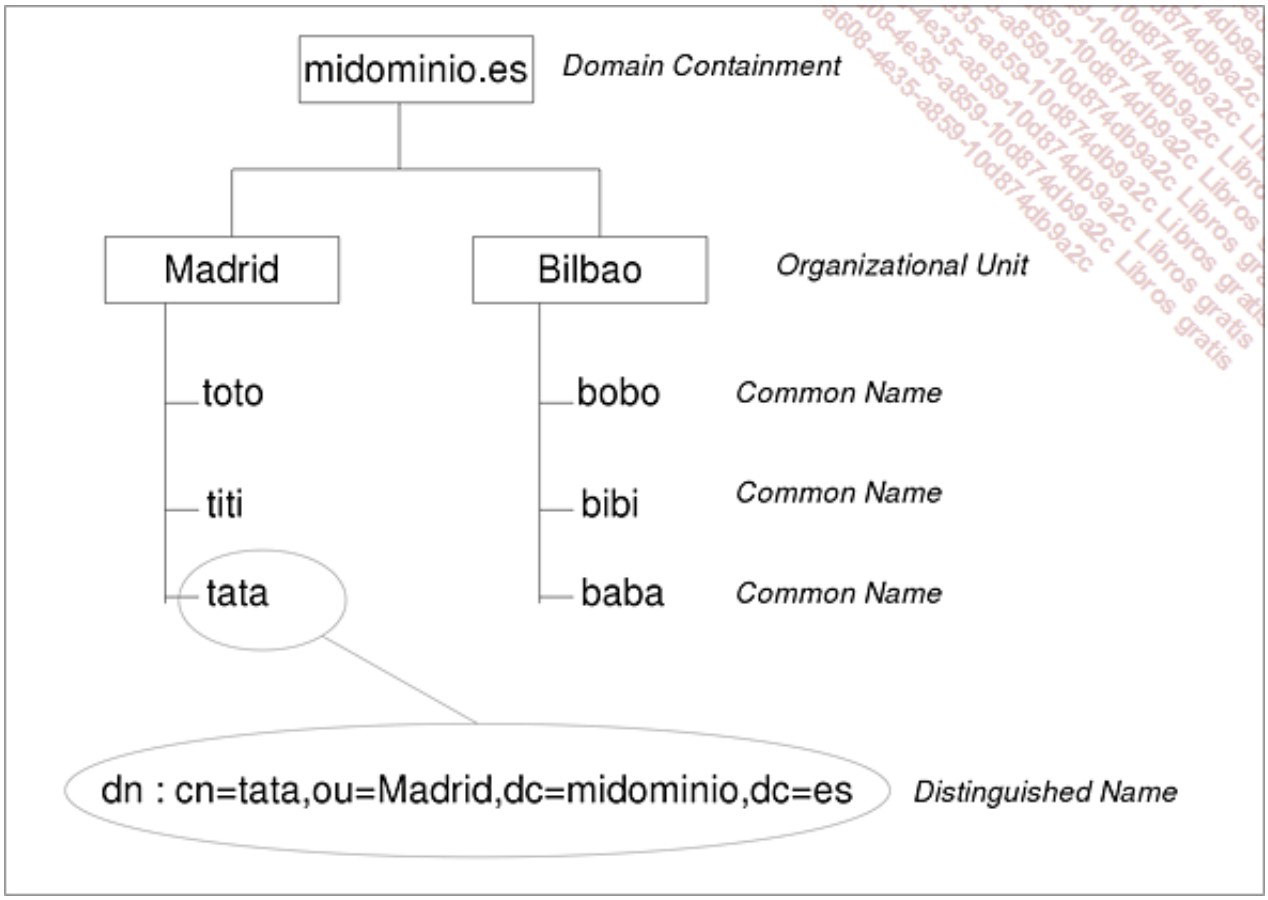
d. El protocolo LDAP

La especificación X500 no previó originalmente que se pudieran realizar consultas ni disponía de un protocolo para tal efecto. En 1993 se realizó un proyecto de protocolo por la universidad de Michigan para crear uno que, funcionando sobre TCP/IP, proporcionara consultas sencillas a un directorio X500: fue el nacimiento de LDAP (*Lightweight Directory Access Protocol*). Los directorios X500 existentes tuvieron que implementar, por tanto, una capa servidor para el protocolo LDAP que permitiera responder a las peticiones de los clientes que usaban este nuevo protocolo.

Rápidamente, el éxito del protocolo LDAP fue tal que se olvidó la función original de X500 para sólo hablar de directorios LDAP. Incluso hoy en día se habla de directorios LDAP para cualquier directorio capaz de responder a peticiones LDAP. Sin embargo, los elementos de estructura y denominación X500 han resistido el paso del tiempo y siempre se habla de objetos, contenedores y esquema.

e. Denominación de objetos

Se ha visto que los objetos del directorio se insertan en una estructura de árbol. Para una denominación sin ambigüedades, existe una notación formal que se basa en la posición del objeto dentro del árbol del directorio activo y en su tipo. Esta notación es el DN (*Distinguished Name*).



Formato típico de un nombre distinguido

```
clase1=nombre_objeto1,clase2=nombre_objeto2,...,clasen=nombre_objeton
```

Donde los parámetros *clasex* representan la clase del objeto descrito (cn, ou, uid, etc.) y los parámetros *objetox* representan los nombres de los objetos descritos.

El nombre distinguido se basa en toda la ramificación del objeto al que se hace referencia hasta la raíz del directorio, cada cambio de nivel se representa mediante comas. Para cada objeto citado, se dice obligatoriamente la clase de este objeto.

El nombre distinguido se utiliza para identificar un objeto del directorio y su uso es obligatorio en operaciones de autentificación.

f. Autentificación con directorios LDAP

Los directorios gestionan su propia seguridad. Aunque a menudo se permiten las peticiones anónimas para consultas en modo lectura, habrá que autentificarse con el directorio para las operaciones de escritura. Esta autentificación se realiza proporcionando el nombre distinguido y la contraseña de una cuenta del directorio con los permisos necesarios en los elementos que se desean gestionar. En terminología LDAP, se habla de "bind" (asociación) para referirse a la autentificación.

g. El formato LDIF

LDIF (*LDAP Data Interchange Format* - Formato de intercambio de datos LDAP) tiene como objetivo permitir la exportación o la importación de datos desde o hacia un directorio LDAP. LDIF describe un formato de archivo de texto que contiene la totalidad o parte de los datos de un directorio LDAP. Puede albergar la totalidad de los objetos y de sus atributos o solamente una selección. Muchas utilidades LDAP usan el formato LDIF.

Formato típico de una entrada de archivo LDIF

```
dn: nombre_distinguido
atributo1: valor1
atributo2: valor2
...
atributon: valorn
```

Es tentador considerar a LDIF como el formato preferido para intercambiar datos de un directorio a otro, en los casos de migración o de intercambio de datos. Los archivos LDIF describen los objetos de un directorio en conformidad a su esquema, pero es muy raro que dos directorios distintos presenten el mismo esquema. Por estos motivos, generalmente el formato LDIF sólo se usa para manipular datos de un mismo directorio, en el caso de una copia de seguridad por ejemplo. Las soluciones de metadirectorios que permiten este tipo de sincronización usan generalmente un formato más abierto como el formato XML.

2. El servidor OpenLDAP

OpenLDAP es la implementación del servidor LDAP open source más común en sistemas Linux. Aunque su facilidad de uso brilla por su ausencia en comparación a sus equivalentes comerciales, no está menos extendido que éstos en todo tipo de implementaciones que van desde la centralización de la autentificación a la gestión de cuentas y libretas de direcciones de correo electrónico.

a. Gestión del servicio

El servicio openldap se administra mediante un script estándar en el directorio **/etc/init.d**. Su nombre es variable y depende de la distribución. La ambigüedad proviene del hecho de que el protocolo aplicativo es LDAP, mientras que el nombre del ejecutable es **slapd** y el nombre del producto aplicativo es openldap.

b. Configuración

En un funcionamiento estándar, tal y como previene la certificación LPI, la configuración inicial no representa una gran cantidad de trabajo. Se trata sobre todo de tener un contexto base: una especie de punto de partida del árbol en el que se encontrarán todos los objetos creados en el directorio. La

configuración se encuentra en el archivo **slapd.conf**, generalmente ubicado en el directorio **/etc/ldap** o **/etc/openldap**. Este archivo también incluye la declaración del administrador del directorio así como su contraseña.

Declaración del contexto base en el archivo slapd.conf

```
suffix      "dc=dominio"
```

Donde *dominio* representa el contexto principal del árbol. Este valor se informa habitualmente en la instalación mediante los scripts posteriores a la instalación de los paquetes. Un mismo directorio openldap puede administrar varios árboles.

Declaración de la cuenta de administrador en el archivo slapd.conf

```
rootdn    "cn=cuenta_admin,dc=dominio"
```

Donde *cuenta_admin* representa la cuenta del administrador del directorio. Atención, a diferencia de otras implementaciones LDAP, no es obligatorio que la cuenta de administrador sea también un objeto del directorio.

Declaración de la contraseña del administrador en el archivo slapd.conf

```
rootpw {método_de_encriptación}contraseña_encriptada
```

Donde *método_de_encriptación* representa el algoritmo de firma utilizado para encriptar la contraseña (SHA1, MD5, crypt o texto sin encriptar).

Para simplificar la introducción de la contraseña, el comando **slappasswd** permite generar la cadena de caracteres formada a partir de la contraseña encriptada con el método de encriptación e insertarla directamente en **slapd.conf**.

Ejemplo de utilización del comando slappasswd

Ya que el comando slappasswd envía su resultado por la salida estándar, hay que ser un poco astuto para integrarlo en el archivo slapd.conf.

```
[root@beta openldap]# slappasswd -s contraseña
{SSHA}oW6wu+yUpFnaB6tg+4cMWnAa8OmDXV62
[root@beta openldap]# echo "rootpw $(slappasswd -s contraseña)" >> slapd.conf
[root@beta openldap]#
```

En este punto, el directorio es completamente funcional después de un reinicio del servicio, pero aún estará vacío. Ya sólo falta llenarlo con los clientes LDAP.

Desde la versión 2.4 de OpenLDAP, la configuración del servidor ya no se almacena en el archivo slapd.conf, sino en el directorio slapd.d que forma parte de una rama del directorio activo. El archivo slapd.conf sigue estando disponible para su uso y puede convertirse en el contexto de una actualización del software.

3. Herramientas LDAP cliente

En Linux hay herramientas por línea de comandos que permiten realizar operaciones en los servidores LDAP. Estas herramientas generalmente las proporciona el paquete **ldap-utils**. Su sintaxis, poco atractiva, requiere un pequeño tiempo de adaptación para usarla con comodidad.

a. Búsqueda de información con ldapsearch

Sin duda, es la herramienta más comúnmente usada de entre las herramientas LDAP cliente por línea de comandos. El comando **ldapsearch** permite efectuar peticiones a un directorio LDAP y recuperar el resultado en formato LDIF.

El caso más simple consiste en solicitar localmente (directamente desde el servidor) la exportación completa de todos los datos de un directorio y a menudo se utiliza esta posibilidad para comprobar la existencia de un objeto o, simplemente, que el directorio responde adecuadamente a las peticiones.

Sintaxis del comando ldapsearch para exportar toda la información pública de un directorio

```
ldapsearch -x -b contexto
```

Sintaxis del comando

Exportación con ldapsearch: opciones y parámetros	
-x	Utilizar una autentificación simple (caso general).
-b *contexto*	Realizar la búsqueda a partir del DN del contenedor del contexto.

ldapsearch para obtener información detallada según unos criterios de búsqueda

```
ldapsearch -x -D dn_admin -W -h ip_servidor -b contexto -s sub atributo=valor
```

Ejemplos de búsqueda con

Búsqueda con ldapsearch: opciones y parámetros	
-D*dn_admin*	Realiza la autentificación con el nombre distinguido *dn_admin*.
-W	Solicitar interactivamente la contraseña. Puede reemplazarse por -w (minúscula) seguido de la contraseña sin encriptar en la línea de comandos.
-h*ip_servidor*	El comando va dirigido al servidor cuya dirección es *ip_servidor*.
-s sub	Realiza una búsqueda recursiva en todos los niveles por debajo del contexto de búsqueda.
atributo	El nombre del atributo que será el criterio de búsqueda.
valor	El valor del atributo buscado. El carácter * representa cualquier valor existente.

ldapsearch

Se desea mostrar todos los usuarios que se encuentran en el directorio cuyo número de teléfono empieza por 91.

Ahora se desea mostrar el conjunto de usuarios de la unidad

```
usuario@ubuntu:~$ ldapsearch -x -D cn=admin,dc=pas,dc=net -w password
-h 172.17.7.20 -b dc=pas,dc=net -s sub telephoneNumber=91*
# extended LDIF
#
# LDAPv3
# base <dc=pas,dc=net> with scope subtree
# filter: telephoneNumber=91*
# requesting: ALL
#

# toto, madrid, pas.net
dn: cn=toto,ou=madrid,dc=pas,dc=net
objectClass: person
cn: toto
sn: toto
telephoneNumber: 9123456789

# tutu, madrid, pas.net
dn: cn=tutu,ou=madrid,dc=pas,dc=net
objectClass: person
cn: tutu
sn: tutu
telephoneNumber: 9178945632

# search result
search: 2
result: 0 Success
```

```
# numResponses: 3
# numEntries: 2
```

organizativa Sevilla. Observe el contexto de búsqueda (-b ou=sevilla,dc=pas,dc=net) y el filtro de búsqueda que intenta comprobar que el atributo teléfono está informado (telephoneNumber=).*

```
usuario@ubuntu:~$ ldapsearch -x -D cn=admin,dc=pas,dc=net -w password
-h 172.17.7.20 -b ou=sevilla,dc=pas,dc=net -s sub telephoneNumber=*
# extended LDIF
#
# LDAPv3
# base <ou=sevilla,dc=pas,dc=net> with scope subtree
# filter: telephoneNumber=*
# requesting: ALL
#

# lolo, sevilla, pas.net
dn: cn=lolo,ou=sevilla,dc=pas,dc=net
objectClass: person
cn: lolo
sn: lolo
telephoneNumber: 9576543210

# lala, sevilla, pas.net
dn: cn=lala,ou=sevilla,dc=pas,dc=net
objectClass: person
cn: lala
sn: lala
telephoneNumber: 9578945632

# search result
search: 2
result: 0 Success

# numResponses: 3
# numEntries: 2
```

Todas las conexiones con el servidor LDAP se realizan con la opción -x, lo cual indica una autentificación en modo texto. Esto, naturalmente, constituye un riesgo en cuanto a la seguridad. La conexión con autentificación SASL permitiría solucionar esta situación. Sin embargo, la complejidad de implantación y el hecho de que la mayor parte de las consultas se realizan de forma anónima hacen que raramente se use la autentificación SASL.

b. Agregar objetos en un directorio con ldapadd

Básicamente, el comando **ldapadd** lee el contenido de un archivo LDIF que contiene los datos que se modificarán y los añade al directorio. La construcción del archivo debe ser rigurosa, pero no presenta dificultad alguna.

Sintaxis simplificada del comando ldapadd

```
ldapadd -x -D dn_admin -W -h ip_servidor -f archivo_ldif
```

ldappadd: opciones y parámetros	
-x	Utilizar una autentificación simple (caso general).
-D*dn_admin*	Realiza la autentificación con el nombre distinguido *dn_admin*.
-W	Solicitar interactivamente la contraseña. Puede reemplazarse por -w (minúscula) seguido de la contraseña sin encriptar en la línea de comandos.

Ejemplo de archivo LDIF para añadir

-h*ip_servidor*	El comando va dirigido al servidor cuya dirección es *ip_servidor*.
-f*archivo_ldif*	Añade los objetos especificados en el archivo *archivo_ldif*.

elementos mediante el comando ldapadd

Llamaremos a este archivo toto.ldif.

Ejemplo de uso de ldapadd

```
dn: cn=toto,dc=pas,dc=net
objectClass: person
cn: toto
sn: toto
telephoneNumber: 9123456789
```

```
root@servidor# ldapadd -D cn=admin,dc=pas,dc=net -W -h 192.168.1.10 -f toto.ldif
root@servidor#
```

c. Modificación de objetos existentes con ldapmodify

El comando **ldapmodify** también se usa con un archivo ldif como argumento y sus parámetros de uso son los mismos que los del comando **ldapadd**.

Sintaxis simplificada del comando ldapmodify

```
ldapmodify -D dn_admin -W -h ip_servidor -f archivo_ldif
```

Ejemplo de archivo LDIF para modificar mediante el comando ldapmodify

```
dn: cn=toto,dc=pas,dc=net
changetype: modify
replace: telephoneNumber
telephoneNumber: 912223344
```

d. Eliminación de objetos con ldapdelete

El comando **ldapdelete** puede emplearse directamente sin usar un archivo ldif.

Ejemplo de eliminación de objetos ldapdelete

```
root@servidor# ldapdelete -D cn=admin,dc=pas,dc=net -w password -h
127.0.0.1 -x cn=toto,dc=pas,dc=net
root@servidor#
```

e. Modificación de contraseñas con ldappasswd

El comando **ldappasswd** permite asignar una contraseña encriptada a un objeto usuario existente en el directorio.

Sintaxis simplificada del comando ldappasswd

```
ldappasswd -x -D dn_admin -W -h ip_servidor -s contraseña dn_usuario
```

Ejemplo de uso del comando

ldappasswd: opciones y parámetro	
-s*contraseña*	La contraseña que se desea asignar al nuevo usuario. Puede reemplazarse por -S (mayúscula) para introducir interactivamente la

	nueva contraseña.
dn_usuario	El nombre distinguido del usuario al que se le quiere modificar la contraseña.

ldappasswd

El primer comando asigna la contraseña al usuario tata. Observe que el uso de las opciones -w y -s permiten incluir las contraseñas (contraseña de autentificación y contraseña del usuario) directamente en la línea de comandos sin tener que introducirlos de forma interactiva.

El segundo comando provoca la visualización de todas las propiedades del usuario tata y se puede ver la contraseña encriptada en el atributo userPassword.

```
usuario@ubuntu:~$ ldappasswd -x -D cn=admin,dc=pas,dc=net -w password
-h 172.17.7.20 -s contraseña cn=tata,ou=madrid,dc=pas,dc=net
usuario@ubuntu:~$ ldapsearch -x -D cn=admin,dc=pas,dc=net -w password
-h 172.17.7.20 -s sub -b dc=pas,dc=net cn=tata
# extended LDIF
#
# LDAPv3
# base <dc=pas,dc=net> with scope subtree
# filter: cn=tata
# requesting: ALL
#

# tata, madrid, pas.net
dn: cn=tata,ou=madrid,dc=pas,dc=net
objectClass: person
cn: tata
sn: tata
telephoneNumber: 9176543210
userPassword:: e1NTSEF9RVpNNVV6RFN1M2xKbUgwZVhDTmpVWGhacEtSOTNxSFU=

# search result
search: 2
result: 0 Success

# numResponses: 2
# numEntries: 1
usuario@ubuntu:~$
```

f. Relajación de la sintaxis para las utilidades LDAP cliente

Cada una de las utilidades por línea de comandos puede encontrar algunos elementos de configuración en el archivo **ldap.conf**. La sintaxis de los comandos se aligerará mucho. Su ubicación generalmente es **/etc/ldap/ldap.conf**, pero puede variar según la implementación.

Archivo ldap.conf común

```
BASE contexto
HOST ip_servidor
```

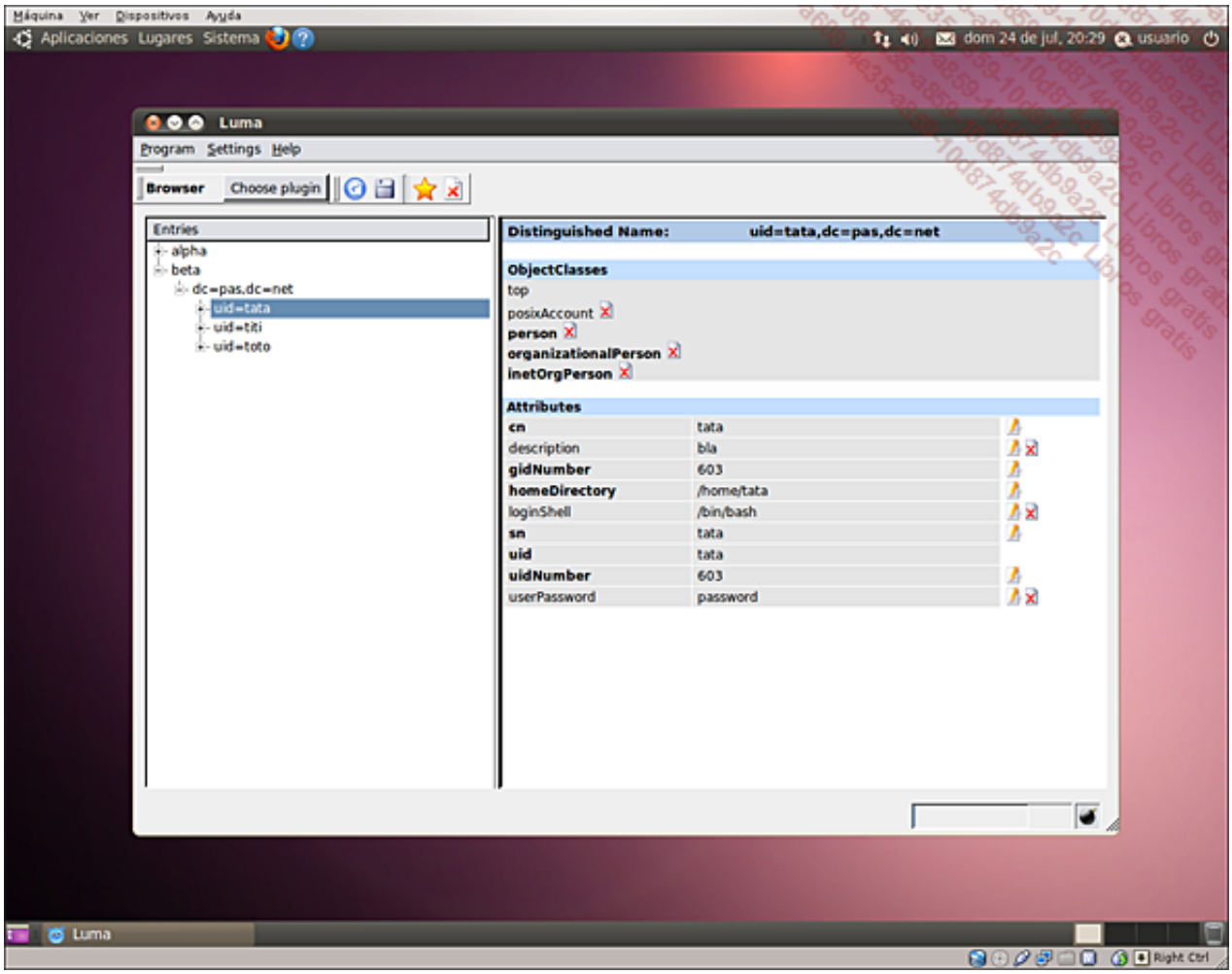
Archivo ldap.conf: parámetros principales	
BASE *contexto*	Realiza las búsquedas a partir del DN del contenedor *contexto*.
HOST*ip_servidor*	Las peticiones se dirigen al servidor cuya dirección es *ip_servidor*.

También se puede declarar el contexto base LDAP mediante la variable LDAPBASE. Sin embargo, el método más universal es informar el archivo ldap.conf.

g. Clientes gráficos

Las aplicaciones compatibles LDAP que integran un cliente les permite realizar peticiones de directorio para su funcionamiento. Por ejemplo, un cliente de mensajería en general es capaz de comprobar la validez de una cuenta o de realizar una búsqueda en el directorio LDAP. Sin embargo, si se utiliza un directorio LDAP al servicio de una aplicación, a menudo es práctico disponer de una utilidad gráfica "universal" que permita comprobar el buen funcionamiento del directorio y si fuera necesario realizar modificaciones independientemente de la aplicación cliente. Son muchas y de calidad variable las herramientas de este tipo. Podemos citar luma, gq y lat.

Ejemplo de visualización desde el cliente gráfico luma



Autentificación por LDAP en sistemas Linux

En el marco de los objetivos LPI, supondremos aquí que ya tenemos un directorio activo en línea y que las cuentas se han creado con todos los atributos necesarios para la autentificación Linux.

1. Configuración NSS

La autentificación sólo será posible si la información de los usuarios está accesible vía NSS.

a. Configuración de la librería NSS para LDAP

La librería NSS, responsable de consultar al directorio, tiene que disponer de la información necesaria. Para ello, hay que rellenar el archivo de configuración LDAP para la librería **nss ldap**. Generalmente, este archivo se llama **ldap.conf** y se ubica directamente en el directorio **/etc**.

Esta configuración requiere que la librería NSS sea capaz de gestionar la información LDAP. Generalmente, esta funcionalidad la proporciona un paquete llamado **libnss_ldap**.

Ejemplo de archivo /etc/dap.conf

Este archivo utilizado por NSS es extremadamente parecido al utilizado por los clientes LDAP.

```
host 127.0.0.1
base dc=pas,dc=net
ldap_version 3
rootbinddn cn=admin,dc=pas,dc=net
```

b. Informando las fuentes de nombres

El archivo **/etc/nsswitch** se debe configurar para hacer referencia a LDAP como fuente de información prioritaria. Sin embargo, debe seguir funcionando con los archivos locales para cuando el directorio no esté disponible.

Modificación del archivo nsswitch con LDAP como fuente de nombres prioritaria

```
passwd : ldap files
group: ldap files
shadow: ldap files
```

c. Comprobación de las fuentes de nombres

A menudo es difícil diagnosticar problemas relacionados con la autentificación LDAP. En efecto, se puede alterar el buen funcionamiento por falta de disponibilidad del directorio, por cuentas de usuarios mal creadas o por una mala configuración del cliente. La herramienta **getent** permite en este caso comprobar que el cliente es capaz de realizar consultas al directorio LDAP y obtener información correcta.

Ejemplo de comprobación de datos de cuentas con getent

La configuración de una autentificación LDAP no es particularmente fácil, esta comprobación en mitad del proceso es bienvenida.

```
root@estacion:/etc$ getent passwd titi
titi:*:1101:1101:titi:/home/titi/bin/bash
root@estacion:/etc$
```

2. Configuración PAM

a. Identificación de los servicios necesarios

Según las necesidades, todos o parte de los servicios que usan PAM tienen que poder apoyarse en la autentificación LDAP. Por ejemplo, las aplicaciones de login, su y ssh solamente para necesidades administrativas, o bien todo elemento capaz de solicitar una autentificación. Siguiendo la filosofía de PAM, habría que identificar todos los elementos de configuración para cada una de las aplicaciones involucradas y modificar su configuración para que puedan usar LDAP como posible mecanismo de autentificación.

Afortunadamente, las distribuciones Linux modernas facilitan la tarea concentrando en los archivos**common-*action*** para Debian o **system-auth** para Red Hat la configuración de todas las aplicaciones que comparten los mismos modos de autentificación. Por tanto, nos bastará con modificar estos archivos para cambiar el modo de autentificación de todas las aplicaciones del sistema.

b. Configuración de los archivos pam

Los tipos de acción PAM **account** y **auth** deben modificarse para permitir la autentificación LDAP. Si se consulta su contenido inicial, se ve que configuran el módulo **pam_unix.so**, en general con el control **required** o **sufficient**. La primera regla es no tocar estas líneas de la configuración. En efecto, incluso si se desea utilizar un directorio LDAP para las operaciones de autentificación, ante todo se debe conservar el mecanismo tradicional, aunque sólo sea para permitir la autentificación local en caso de fallo del directorio. Por tanto, la configuración se realizará añadiendo para las acciones **account** y **auth** una línea indicando como **sufficient** la autenticación por el módulo LDAP (pam_ldap.so). Se podrá evitar una doble entrada de contraseña añadiendo la opción**use_first_pass** que permitirá reutilizar la contraseña introducida en el primer intento de conexión.

Extracto del archivo system-auth modificado en una distribución Red Hat

El parámetro use_first_pass indica al sistema que debe intentar la autentificación en el módulo pam_ldap con los mismos identificadores que los que se usaron con el módulo pam_unix. Así se evita que el usuario introduzca dos veces sus credenciales.

```
auth     sufficient  pam_unix.so  nullok
auth     sufficient  pam_ldap.so  use_first_pass
account  sufficient  pam_unix.so
account  sufficient  pam_ldap.so
```

Comprobación de los conocimientos adquiridos: preguntas/respuestas

Ponga a prueba sus conocimientos respondiendo a las preguntas siguientes. Estas preguntas no siempre esperan una respuesta cerrada. Las preguntas planteadas en la certificación, a pesar de que abordan los mismos temas, en su mayoría serán planteadas en forma de preguntas de opción múltiple o que requieran una respuesta corta, en palabras escritas en el teclado.

1. Preguntas

1. ¿Por qué las contraseñas de los sistemas actuales ya no se almacenan en el archivo /etc/passwd como se hacía en los sistemas Unix originales?
2. ¿Por qué hay un parámetros dns en el archivo /etc/nsswitch.conf?
3. ¿En qué aspecto la aparición de PAM ha facilitado el trabajo de los desarrolladores en lo que respecta a las operaciones de autentificación?
4. ¿Cuál es el interés del concepto de pila de módulos PAM?
5. ¿En qué caso un módulo de autentificación PAM llamado con el control de comportamiento sufficient no conduce al éxito de la autentificación?
6. ¿Qué sucede después del éxito de un modulo llamado con el control de comportamiento required?
7. ¿Cómo se puede utilizar el formato de intercambio LDIF de los directorios LDAP para exportar los datos de un directorio LDAP como Active Directory a otro directorio LDAP como OpenLDAP?
8. ¿Por qué es necesario un comando específico (ldappasswd) para modificar la contraseña de una cuenta de usuario openldap cuando el comando ldapmodify ya permite escribir cualquier atributo de objeto en el directorio?
9. ¿Existe algún método concreto que permita definir el contexto de búsqueda de los clientes LDAP distinto al de informar la directiva BASE en el archivo ldap.conf?
10. ¿Por qué en una autentificación LDAP de un sistema Linux se conserva casi siempre la autentificación local mediante los archivos de contraseña (/etc/shadow)?

2. Respuestas

1. ¿Por qué las contraseñas de los sistemas actuales ya no se almacenan en el archivo /etc/passwd como se hacía en los sistemas Unix originales?

   Las contraseñas originalmente se almacenaban en el archivo /etc/passwd con el resto de información relacionada con las cuentas de los usuarios. Este archivo debía de estar accesible en modo lectura para todos los usuarios y las contraseñas encriptadas mediante un algoritmo de hash. Con la evolución de la potencia de cálculo de los ordenadores se hizo posible poder adivinar una contraseña encriptando en primer lugar todas las entradas de un diccionario y pasando después todas las combinaciones de caracteres posibles. Encontrar la cadena de caracteres que una vez encriptada fuera la misma que la del archivo suponía encontrar la contraseña. Para evitar esta vulnerabilidad, las contraseñas se retiraron del archivo /etc/passwd y se ubicaron en el archivo /etc/shadow, al que los usuarios no tienen acceso.

2. ¿Por qué hay un parámetros dns en el archivo /etc/nsswitch.conf?

   El archivo nsswitch.conf contiene todos los parámetros de resolución de nombres así como las bases de datos de información necesarias para su resolución. De este modo, generalmente indica que la resolución de nombres de equipos (hosts) debe realizarse en primer lugar mediante un archivo local (parámetro files) y después por un servicio dns si se ha configurado (parámetro dns).

3. ¿En qué aspecto la aparición de PAM ha facilitado el trabajo de los desarrolladores en lo que respecta a las operaciones de autentificación?

   Porque los desarrolladores sólo tienen que hacer sus aplicaciones compatibles con la librería de autentificación PAM. Si las técnicas de autentificación evolucionan, no será necesario modificar la aplicación, sino que lo único que habrá que modificar es su configuración PAM que determinará los nuevos módulos en los que se basa esta autentificación.

4. ¿Cuál es el interés del concepto de pila de módulos PAM?

El interés de poder englobar en la operación de autentificación la ejecución de varios módulos. Las aplicaciones prácticas son muchas: se puede aceptar un usuario si se produce con éxito una autentificación remota (LDAP) o si la autentificación local tiene éxito. Se puede exigir que una autentificación particular (biométrica por ejemplo) tenga éxito, así como una autentificación tradicional mediante contraseña. Finalmente, algo que también se hace a menudo es aprovechar la etapa de autentificación para ejecutar otras acciones como cargar variables (módulo pam_env) o prohibir ciertos accesos de usuarios (módulo pam_nologin).

5 ¿En qué caso un módulo de autentificación PAM llamado con el control de comportamiento sufficient no conduce al éxito de la autentificación?

Si un módulo llamado con el control de comportamiento required o requisite ha fallado previamente.

6 ¿Qué sucede después del éxito de un modulo llamado con el control de comportamiento required?

Se continua. El éxito de un módulo required no provoca la parada del tratamiento de la pila. Los otros módulos de la pila se ejecutarán.

7 ¿Cómo se puede utilizar el formato de intercambio LDIF de los directorios LDAP para exportar los datos de un directorio LDAP como Active Directory a otro directorio LDAP como OpenLDAP?

Es muy difícil. El formato LDIF está íntimamente ligado al esquema del directorio y dos directorios distintos tienen casi siempre distintos esquemas. Los atributos LDAP, por consiguiente, no serán los mismos por ambos lados y una exportación contendrá seguramente elementos no asimilables por el segundo directorio. Se podría puntualmente tener éxito en algunos intercambios restringiendo los datos exportados e importados a clases de objetos y atributos comunes en los dos directorios. Los servicios funcionales que permiten intercambios completos (metadirectorios) se basan siempre en una fase de reformateo de datos. A menudo se habla de una operación de mapeo de atributos.

8 ¿Por qué es necesario un comando específico (ldappasswd) para modificar la contraseña de una cuenta de usuario openldap cuando el comando ldapmodify ya permite escribir cualquier atributo de objeto en el directorio?

Porque el comando ldapmodify escribiría el atributo de la contraseña sin encriptar, mientras que el comando ldappasswd gestiona de forma nativa varios algoritmos de encriptación.

9¿Existe algún método concreto que permita definir el contexto de búsqueda de los clientes LDAP distinto al de informar la directiva BASE en el archivo ldap.conf?

Sí, se puede informar la variable LDAPBASE con el contexto de búsqueda que deberán usar los clientes LDAP. Sin embargo, este método sufre el problema de la volatilidad de las variables y la declaración deberá guardarse en el entorno de ejecución de los comandos cliente (la variable se exporta generalmente desde un proceso padre de los comandos cliente).

10 ¿Por qué en una autentificación LDAP de un sistema Linux se conserva casi siempre la autentificación local mediante los archivos de contraseña (/etc/shadow)?

Para preservar el uso del sistema en caso que el directorio LDAP no esté disponible. Los controles de comportamiento de los módulos LDAP se adaptan a este uso permitiendo la autentificación por directorio LDAP pero valiéndose de un método alternativo en caso de error.

Trabajos prácticos

Ante las perspectivas de crecimiento de la empresa y los miles de puestos de trabajo posibles, se va dando cuenta de la necesidad de una autentificación centralizada y a gran escala. Por tanto, decide instalar un servidor LDAP en beta.

1. Creación y alimentación de un directorio LDAP en el servidor beta

a. Instalación de los paquetes de software

En el servidor **beta**, instale el servicio LDAP así como las utilidades cliente mediante el comando siguiente:

```
yum install openldap-servers
yum install openldap-clients
```

En la estación de trabajo, instale las utilidades cliente mediante el comando siguiente:

```
sudo apt-get install ldap-utils
```

b. Configuración del directorio

Archivos y comandos útiles

- /etc/openldap/slapd.conf
- rootpw
- slaptest
- vi

Operaciones

Estas operaciones se realizan con la versión 2.4 del servidor openldap en CentOS 6. Tienen en cuenta la reciente modificación de configuración del servidor y el abandono del archivo slapd.conf. Por razones pedagógicas y para encuadrar mejor los objetivos LPI, utilizaremos el archivo ”antiguo” y lo convertiremos al nuevo formato con el comando slaptest. Este comando sirve originalmente para comprobar la coherencia de un archivo de configuración, pero también es capaz de convertir el archivo de configuración histórico en la estructura de carpetas necesaria para el nuevo formato de configuración (slapd.d).

1. En el servidor beta, localice el archivo **slapd.conf** de ejemplo y cópielo en el directorio **/etc/openldap** quitándole cualquier extensión innecesaria.
2. En el archivo **slapd.conf**, quite la declaración de contexto por defecto ”my-domain.com” (dc=my-domain, dc=com) y reemplácela por ”pas.net” (dc=pas, dc=net). No olvide modificar el sufijo del rootdn.
3. En el archivo **slapd.conf**, directamente en la sección **database config**, informe la contraseña del administrador con el valor ”password”.
4. En el archivo **slapd.conf**, en la sección **database dbd** y justo debajo de la directiva**rootdn**, informe la contraseña del administrador con el valor ”password”.
5. Elimine el contenido del directorio **/etc/openldap/slapd.d**.
6. Convierta su archivo de configuración con el comando **slaptest -f slapd.conf -F slapd.d**. Ignore por ahora los mensajes de error y compruebe que se crea nuevo contenido en el directorio **slapd.d**.
7. Asigne a la cuenta y al grupo de servicio ldap la propiedad del directorio**/var/lib/ldap**.
8. Reasigne el contenido del directorio slapd a la cuenta y grupo del servicio ldap.
9. Inicie el servicio.

Resumen de los comandos y resultado por pantalla

Comprobación del reemplazo del sufijo:

```
[root@beta openldap]# grep "dc=pas,dc=net" slapd.conf
        by dn.exact="cn=Manager,dc=pas,dc=net" read
suffix          "dc=pas,dc=net"
rootdn          "cn=Manager,dc=pas,dc=net"
```

Comprobación de la presencia de las dos entradas rootpw:

Archivo slapd.conf

```
[root@beta openldap]#
```

después de la modificación (sin los comentarios):

```
[root@beta openldap]# grep rootpw slapd.conf
rootpw          password
rootpw          password
# rootpw        {crypt}ijFYNcSNctBYg
[root@beta openldap]#
```

Borramos el contenido de slapd.d:

Recreación del contenido de slapd.d:

```
include         /etc/openldap/schema/corba.schema
include         /etc/openldap/schema/core.schema
include         /etc/openldap/schema/cosine.schema
include         /etc/openldap/schema/duaconf.schema
include         /etc/openldap/schema/dyngroup.schema
include         /etc/openldap/schema/inetorgperson.schema
include         /etc/openldap/schema/java.schema
include         /etc/openldap/schema/misc.schema
include         /etc/openldap/schema/nis.schema
include         /etc/openldap/schema/openldap.schema
include         /etc/openldap/schema/ppolicy.schema
include         /etc/openldap/schema/collective.schema

allow bind_v2


pidfile         /var/run/openldap/slapd.pid
argsfile        /var/run/openldap/slapd.args



TLSCACertificatePath /etc/openldap/certs
TLSCertificateFile "\"OpenLDAP Server\""
TLSCertificateKeyFile /etc/openldap/certs/password



database config
access to *
        by dn.exact="gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth"
manage
        by * none
rootpw  password

database monitor
access to *
        by dn.exact="gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth"
read
        by dn.exact="cn=Manager,dc=pas,dc=net" read
        by * none


database        bdb
suffix          "dc=pas,dc=net"
checkpoint      1024 15
rootdn          "cn=Manager,dc=pas,dc=net"
rootpw  password

directory       /var/lib/ldap

index objectClass                       eq,pres
index ou,cn,mail,surname,givenname      eq,pres,sub
index uidNumber,gidNumber,loginShell    eq,pres
index uid,memberUid                     eq,pres,sub
index nisMapName,nisMapEntry            eq,pres,sub
```

Asignación de archivos ldap a la cuenta y grupo del servicio:

```
[root@beta openldap]# ls
cacerts  ldap.conf  schema  slapd.conf  slapd.d
```

Parada e inicio del servicio:

c. Consulta sencilla al directorio

Un directorio openldap es por definición muy discreto. En este punto, usted decide realizar una consulta sencilla al directorio para ver si va a responder.

Comandos útiles

```
[root@beta openldap]# rm -rf slapd.d/*
[root@beta openldap]#
```

```
[root@beta openldap]# slaptest -f slapd.conf -F slapd.d
bdb_db_open: warning - no DB_CONFIG file found in directory /var/lib/ldap: (2).
Expect poor performance for suffix "dc=pas,dc=net".
bdb_db_open: database "dc=pas,dc=net": db_open(/var/lib/ldap/id2entry.bdb)
failed: No such file or directory (2).
backend_startup_one (type=bdb, suffix="dc=pas,dc=net"): bi_db_open failed! (2)
slap_startup failed (test would succeed using the -u switch)
[root@beta openldap]# ls -l slapd.d
total 8
drwxr-x---. 3 root root 4096 oct  1 18:04 cn=config
-rw-------. 1 root root 1099 oct  1 18:04 cn=config.ldif
[root@beta openldap]#
```

```
[root@beta openldap]# ls -l slapd.d
total 8
drwxr-x---. 3 root root 4096 oct  1 18:04 cn=config
-rw-------. 1 root root 1099 oct  1 18:04 cn=config.ldif
[root@beta openldap]# chown -R ldap:ldap slapd.d
[root@beta openldap]# ls -l slapd.d
total 8
drwxr-x---. 3 ldap ldap 4096 oct  1 18:04 cn=config
-rw-------. 1 ldap ldap 1099 oct  1 18:04 cn=config.ldif
[root@beta openldap]# ls -l /var/lib/ldap
total 576
-rw-r--r--. 1 root root   2048 oct  1 18:04 alock
-rw-------. 1 root root  24576 oct  1 18:04 __db.001
-rw-------. 1 root root 147456 oct  1 18:04 __db.002
-rw-------. 1 root root 270336 oct  1 18:04 __db.003
-rw-------. 1 root root  98304 oct  1 18:04 __db.004
-rw-------. 1 root root 491520 oct  1 18:04 __db.005
-rw-------. 1 root root  32768 oct  1 18:04 __db.006
[root@beta openldap]# chown -R ldap:ldap /var/lib/ldap
[root@beta openldap]# ls -l /var/lib/ldap
total 576
-rw-r--r--. 1 ldap ldap   2048 oct  1 18:04 alock
-rw-------. 1 ldap ldap  24576 oct  1 18:04 __db.001
-rw-------. 1 ldap ldap 147456 oct  1 18:04 __db.002
-rw-------. 1 ldap ldap 270336 oct  1 18:04 __db.003
-rw-------. 1 ldap ldap  98304 oct  1 18:04 __db.004
-rw-------. 1 ldap ldap 491520 oct  1 18:04 __db.005
-rw-------. 1 ldap ldap  32768 oct  1 18:04 __db.006
[root@beta openldap]#
```

```
[root@beta openldap]# service slapd stop
Parando slapd:                                             [FALLÓ]
[root@beta openldap]# service slapd start
Iniciando slapd:                                           [  OK  ]
[root@beta openldap]#
```

ldapsearch

- pgrep
- service

Operaciones

1. En el servidor **beta**, comprobar que el servicio slapd se está ejecutando.
2. Desde el servidor **beta**, hacer una petición lo más simple posible con el objetivo de obtener una respuesta del directorio (aunque en este momento el directorio esté vacío).

Resumen de los comandos y resultado por pantalla

Comprobación de la ejecución del servicio por dos métodos distintos:

```
[root@beta openldap]# pgrep -l slapd
2959 slapd
[root@beta openldap]# service ldap status
Se está ejecutando slapd (pid 2959)...
[root@beta openldap]#
```

```
[root@beta openldap]# ldapsearch -x
# extended LDIF
#
# LDAPv3
# base <> with scope subtree
# filter: (objectclass=*)
# requesting: ALL
#

# search result
search: 2
result: 32 No such object

# numResponses: 1
[root@beta openldap]#
```

Petición simple:

d. Creación de un contexto base

Comandos útiles

- ldappadd
- vi

Operaciones

1. Crear un archivo LDIF que contenga la declaración del contexto base.
2. Importar este archivo en el directorio.

```
dc: pas
```

Archivo base.ldif:

```
dn: dc=pas, dc=net
objectClass: domain
```

Resumen de los comandos y resultado por pantalla

Comprobación del archivo base.ldif:

```
[root@beta ~]# cat base.ldif
dn: dc=pas, dc=net
objectClass: domain
dc: pas
[root@beta ~]#
```

```
[root@beta openldap]# ldapadd -x -D cn=Manager,dc=pas,dc=net -W -f ~/base.ldif
Enter LDAP Password :
adding new entry "dc=pas,dc=net"
[root@beta openldap]#
```

Importación del archivo en el directorio:

e. Creación de cuentas de usuario

Comandos útiles

- ldappadd
- vi

Operaciones

1. Crear un archivo LDIF que contenga los datos de dos usuarios.
2. Importar este archivo en el directorio.

Archivo usuarios.ldif:

```
dn: uid=toto,dc=pas,
objectClass: top
```

```
objectClass: posixAccount
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
uid: toto
cn: toto
sn: toto
uidNumber: 601
gidNumber: 1000
homeDirectory: /home/toto
loginShell: /bin/bash
userPassword: password

dn: uid=titi,dc=pas,dc=net
objectClass: top
objectClass: posixAccount
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
uid: titi
cn: titi
sn: titi
uidNumber: 602
gidNumber: 1000
homeDirectory: /home/titi
loginShell: /bin/bash
userPassword: password
```

Resumen de los comandos y resultado por pantalla

```
[root@beta openldap]# ldapadd -x -D cn=Manager,dc=pas,dc=net -W -f ~/usuarios.ldif
Enter LDAP Password :
adding new entry "uid=toto,dc=pas,dc=net"
adding new entry "uid=titi,dc=pas,dc=net"
[root@beta openldap]#
```

f. Consulta de un directorio poblado

Comando útil

- ldapsearch

Operaciones

1. Desde el servidor beta, hacer una consulta con el objetivo de obtener la totalidad de los datos del directorio.

Resumen de los comandos y resultado por pantalla

Petición LDAP:

```
[root@beta openldap]# ldapsearch -x -b "dc=pas,dc=net" -D
cn=Manager,dc=pas,dc=net -W -s sub
Enter LDAP Password:
# extended LDIF
#
# LDAPv3
# base <dc=pas,dc=net> with scope subtree
# filter: (objectclass=*)
# requesting: ALL
#

# pas.net
dn: dc=pas,dc=net
objectClass: domain
dc: pas

# toto, pas.net
dn: uid=toto,dc=pas,dc=net
objectClass: top
objectClass: posixAccount
objectClass: person
objectClass: organizationalPerson
```

g. Consulta del directorio desde un cliente

Antes de pasar a temas más serios, nos falta comprobar que el cliente de la estación de trabajo recibe adecuadamente los datos del directorio.

```
objectClass: inetOrgPerson
uid: toto
cn: toto
sn: toto
givenName: toto
uidNumber: 601
gidNumber: 1000
homeDirectory: /home/toto
loginShell: /bin/bash
userPassword:: cGFzc3dvcmQ=
(...)
[root@beta openldap]#
```

Archivos y comandos útiles

- ldap.conf
- ldapsearch

Operaciones

1. Desde la estación de trabajo, hacer una consulta del contenido del directorio detallando todos los elementos necesarios por línea de comandos.
2. Informar los parámetros BASE y HOST en el archivo **/etc/ldap/ldap.conf**.
3. Volver a hacer una consulta en el servidor con una sintaxis aligerada basándose en los datos del archivo **ldap.conf**.

Resumen de los comandos y resultado por pantalla

Petición desde la estación de trabajo:

```
usuario@estacion:~$ ldapsearch -x -h 192.168.200.102 -b dc=pas,dc=net
-D cn=Manager,dc=pas,dc=net -W -s sub
Enter LDAP Password:
# extended LDIF
#
# LDAPv3
# base <dc=pas,dc=net> with scope subtree
# filter: (objectclass=*)
# requesting: ALL
#

# pas.net
dn: dc=pas,dc=net
objectClass: domain
dc: pas

# toto, pas.net
dn: uid=toto,dc=pas,dc=net
(...)
usuario@estacion:~$
```

Archivo /etc/ldap/ldap.conf modificado:

```
#
# LDAP Defaults
#

# See ldap.conf(5) for details
# This file should be world readable but not world writable.

BASE    dc=pas,dc=net
HOST    192.168.200.102
#URI    ldap://ldap.example.com ldap://ldap-master.example.com:666

#SIZELIMIT      12
#TIMELIMIT      15
#DEREF          never
```

Petición aligerada:

2.

```
usuario@estacion:~$ ldapsearch -x -D cn=Manager,dc=pas,dc=net -W -s sub
Enter LDAP Password:
# extended LDIF
#
# LDAPv3
# base <dc=pas,dc=net> with scope subtree
# filter: (objectclass=*)
# requesting: ALL
#

# pas.net
dn: dc=pas,dc=net
objectClass: domain
dc: pas

# toto, pas.net
(...)
usuario@estacion:~$
```

Autentificación del puesto de trabajo mediante el directorio LDAP

a. Instalación de los elementos software necesarios para la autentificación LDAP

En la estación de trabajo estacion, instale las librerías pam necesarias para la autentificación LDAP:

```
sudo apt-get install ldap-auth-client
sudo apt-get install ldap-auth-config
```

Responda a las pocas preguntas que se le plantean. De todos modos, volverá a los archivos de configuración para su configuración. En caso de duda, ponga un valor claramente inapropiado para que sea más fácil de identificar los elementos que habrá informado el asistente.

b. Configuración de la resolución de nombres LDAP

Comandos y archivos útiles

- /etc/ldap.conf
- /etc/nsswitch.conf
- getent

Operaciones

1. En el archivo **nsswitch.conf**, añadir la palabra clave ldap a la secciones passwd, group y shadow.
2. Para que las resoluciones de nombres puedan realizarse por LDAP, introducir los parámetros **host** y **base** en el archivo **/etc/ldap.conf**. Para una mejor estabilidad, comentar o borrar la línea uri ldapi://.
3. Comprobar que la resolución se realiza correctamente y que existe un nombre de usuario asociado correctamente a una cuenta de usuario en el directorio.

Resumen de los comandos y resultado por pantalla

Extracto del archivo /etc/nsswitch.conf modificado:

```
passwd : ldap compat
group :  ldap compat
shadow : ldap compat
```

Extracto del archivo /etc/ldap.conf modificado:

```
host 192.168.200.201
base dc=pas,dc=net
# uri ldapi://192.168.200.201
```

Prueba de la resolución de nombres:

c. Configuración de la autentificación

```
usuario@estacion:/etc$ getent passwd titi
titi:*:602:1000:titi:/home/titi:/bin/bash
usuario@estacion:/etc$
```

pam con LDAP

Sin lugar a dudas, los archivos pam ya han sido modificados por los scripts posteriores a la instalación y la configuración ya debería ser funcional. La distribución Ubuntu, siendo un poco vanguardista, habrá configurado los archivos correspondientes. Vamos a poner a los parámetros pam los valores estándar definidos por la certificación LPI.

Comandos y archivos útiles

- /etc/pam.d/common-account
- /etc/pam.d/common-auth

Operaciones

1. En el archivo **/etc/pam.d/common-auth**, verificar que existe una línea para el módulo **pam_unix** y declararla como **sufficient**.
2. En el archivo **/etc/pam.d/common-auth**, verificar que existe una línea para el módulo **pam_ldap** (o añadirla) y declararla como **sufficient**.
3. En el archivo **/etc/pam.d/common-account**, verificar que existe una línea para el módulo **pam_unix** y declararla como **sufficient**.
4. En el archivo **/etc/pam.d/common-account**, verificar que existe una línea para el módulo **pam_ldap** (o añadirla) y declararla como **sufficient**.
5. Para crear bajo demanda un directorio personal a un nuevo usuario que se conecte, añadir al archivo **/etc/pam.d/common-session** una línea que cargue el módulo **pam_mkhomedir.so** en el tipo **session**, con el control **required** y con la opción **skel=/etc/skel**.

Archivos modificados

Archivo /etc/pam.d/common-auth:

Archivo

```
auth    sufficient      pam_unix.so nullok_secure
auth    sufficient      pam_ldap.so use_first_pass
auth    requisite       pam_deny.so
auth    required        pam_permit.so
auth    optional        pam_cap.so
```

/etc/pam.d/common-account:

Archivo

```
account    sufficient      pam_unix.so
account    sufficient      pam_ldap.so
account    requisite       pam_deny.so
account    required        pam_permit.so
```

/etc/pam.d/common-session:

```
session    [default=1]     pam_permit.so
session    requisite       pam_deny.so
session    required        pam_permit.so
session    optional        pam_umask.so
session    required        pam_unix.so
session    required        pam_mkhomedir.so skel=/etc/skel
session    optional        pam_ldap.so
session    optional        pam_ck_connector.so nox11
```

d. Validación funcional

La estación de trabajo Ubuntu debería ser ahora capaz de abrir una sesión con una cuenta de usuario situada en el directorio LDAP del servidor beta.

Modificar cualquier parámetro pam es peligroso en sí mismo, se recomienda comprobar la configuración con un comando que

no requiera reiniciar el sistema, como el comando su. En caso de error, se dispondrá de todo el tiempo para restablecer la configuración y evitar tener que reinstalar un sistema incapaz de reiniciar.

Requisitos y objetivos

1. Requisitos

Los conocimientos adquiridos en la certificación LPI nivel 1, especialmente:

- ☒ Montaje de sistemas de archivos.
- ☒ Edición de archivos.

2. Objetivos

Al final de este capítulo, será capaz de:

- ☒ Conocer los daemons NFS.
- ☒ Exportar comparticiones NFS puntuales.
- ☒ Configurar un servicio NFS.
- ☒ Conocer los comandos de diagnóstico NFS.
- ☒ Comprender la administración de los derechos de acceso de clientes NFS.
- ☒ Conectar un cliente a una compartición NFS.
- ☒ Conocer los daemons samba.
- ☒ Configurar la compartición samba de los directorios personales de los usuarios.
- ☒ Conocer las opciones samba más comunes.
- ☒ Administrar las contraseñas samba.
- ☒ Conectar un cliente a una compartición samba.
- ☒ Conocer los modos de funcionamiento FTP.
- ☒ Configurar un servidor FTP.
- ☒ Usar un cliente FTP.

Compartición de datos con NFS

NFS es el protocolo tradicional de compartición de archivos en sistemas Unix. Aunque su antigüedad lo hace menos popular a ojos de los jóvenes usuarios de Linux, sigue siendo interesante conocerlo por la rapidez y la simplicidad en su despliegue para una compartición entre dos sistemas Linux o Unix. Además, NFS está gozando recientemente de un renovado interés gracias a algunas aplicaciones que lo usan, como las infraestructuras Vmware para acceder a espacios de almacenamiento de bajo coste o las unidades multimedia domésticas que acceden a servidores de archivos.

1. Compartición de directorios

a. Observación de comparticiones activas

Las comparticiones NFS activas en un sistema se declaran mediante un directorio local y están accesibles para algunos clientes con ciertas opciones. Los clientes autorizados así como las opciones se declaran cuando se activa la compartición. Si se encuentra un sistema ya configurado, puede ser útil hacer un diagnóstico de las comparticiones activas de este sistema. Este diagnóstico se realiza usando el comando **exportfs**.

Ejemplo de uso del comando exportfs para observar las comparticiones activas

En este ejemplo, el directorio /perso se comparte solamente para la dirección 192.168.0.20, mientras que /nas se comparte para todos los clientes.

```
alfa:~# exportfs
/data/perso          <192.168.0.20>
/nas       <world>
alfa:~#
```

Se pueden observar las estadísticas relacionadas con la actividad NFS mediante el comando**nfsstat**.

Visualización de estadísticas de NFS

El comando nfsstat sirve sobre todo para verificar una actividad o una ausencia de actividad en un servidor NFS.

```
usuario@servidor:~$ nfsstat
Server rpc stats:
calls      badcalls   badauth    badclnt    xdrcall
12         0          0          0          0

Server nfs v3:
null         getattr      setattr      lookup       access       readlink
2        18% 2        18% 0         0% 2        18% 1         9% 0         0%
read         write        create       mkdir        symlink      mknod
0         0% 0         0% 0         0% 0         0% 0         0% 0         0%
remove       rmdir        rename       link         readdir      readdirplus
0         0% 0         0% 0         0% 0         0% 0         0% 0         0%
fsstat       fsinfo       pathconf     commit
0         0% 3        27% 1         9% 0         0%

Client rpc stats:
calls      retrans    authrefrsh
0          0          0

usuario@servidor:~$
```

b. Compartición puntual

El comando **exportfs** también permite declarar una compartición de forma interactiva. Se utiliza para declarar comparticiones puntuales.

Sintaxis del comando exportfs para una compartición puntual

```
exportfs dirección_cliente:/ruta_compartición
```

Comando exportfs: opciones y parámetros	
dirección_cliente	Dirección IP del cliente o de la red que se puede conectar a la compartición. El comodín * autoriza a todos los clientes para que se puedan conectar.
ruta_compartición	Ruta absoluta del directorio que se comparte.

Por supuesto, hace ya mucho tiempo que el control de acceso basado en la dirección IP no es una garantía de seguridad.

c. Servicio NFS y compartición permanente

Naturalmente, se puede declarar una compartición permanente activa para el inicio del servicio NFS. Esta declaración se realiza en el archivo **/etc/exports**. Hay que tener en cuenta que, según la distribución, este archivo posiblemente no se haya creado después de la instalación del servicio y hay que crearlo desde cero.

Formato del archivo /etc/exports

```
compartición1 dirección_cliente1
compartición2 dirección_cliente2
```

Este archivo se lee cada vez que arranca el servicio NFS o en cada llamada al comando **exporfs**con la opción **-a**. Hay que tener en cuenta que las comparticiones se expresan siempre con su ruta absoluta, es decir, se expresan en relación a la raíz del sistema de archivos.

El script de administración del servicio NFS provoca la ejecución de tres daemons estándar:

- portmap: gestiona las peticiones RPC (*Remote Procedure Call*).
- nfsd: espacio de usuario del servicio NFS. Inicia los threads NFS para las conexiones cliente.
- mountd: gestiona las peticiones de montaje de los clientes.

El comando **rpcinfo** permite efectuar una petición RPC a un servidor y mostrar los daemons gestionados.

d. Opciones de compartición

Algunas opciones modifican el comportamiento del servidor NFS para cada una de las comparticiones que alberga. Las opciones se especifican mediante el comando **exportfs** si se utiliza dinámicamente o en el archivo **/etc/exports** si se utiliza NFS como servicio.

Opciones NFS comunes	
ro	Acceso en modo sólo lectura.
rw	Acceso en modo lectura/escritura.
sync	Acceso en modo escritura síncrona. Los datos se escriben inmediatamente.
async	Acceso en modo escritura asíncrona. Utilización de una caché de escritura.
root_squash	Comportamiento por defecto. La cuenta root pierde sus privilegios en la compartición.
no_root_squash	La cuenta root conserva sus privilegios en la compartición.
nolock	No se bloquean los archivos a los que se accede.

Ejemplo de uso del comando exportfs con la opción de sólo lectura

Si hay varias opciones, se deben separar por comas.

```
root@servidor# exportfs -o ro *:/data
```

Ejemplo de archivo

```
root@servidor#
```

/etc/exports con la opción de sólo lectura

*El parámetro * o una dirección IP de un cliente autorizado son indispensables para un buen funcionamiento.*

Ejemplo de

```
/data      *(ro)
```

visualización de las comparticiones activas con sus opciones

Se muestran las opciones explícitas así como las opciones por defecto.

```
alfa:~# exportfs -v
/perso          192.168.0.20(rw,wdelay,root_squash,no_subtree_check)
/data           <world>(ro,wdelay,root_squash,no_subtree_check)
alfa:~#
```

2. Configuración de clientes

a. Visualización de las comparticiones remotas

El comando **showmount** permite mostrar la información de un servidor NFS remoto.

Visualización de las comparticiones remotas con showmount

```
showmount --exports servidor
```

Donde *servidor* representa la dirección IP del servidor del que queremos obtener las comparticiones.

b. Montaje de un directorio remoto

Los ordenadores cliente acceden a una compartición NFS mediante una operación de montaje. Usan la compartición montada como si de una estructura de directorios local se tratase.

Montaje de una compartición NFS

```
mount -t nfs dirección_servidor:/ruta_compartición punto_de_montaje
```

Montaje NFS: opciones y parámetros	
-t nfs	Indica que el dispositivo que se montará es una compartición NFS remota e invoca al subprograma cliente NFS.
dirección_servidor	La dirección IP del servidor NFS.
ruta_compartición	La ruta absoluta del directorio compartido en el servidor.
punto_de_montaje	El directorio local del cliente en el que se montará la compartición NFS.

3. Administración de las identidades

a. Los permisos del cliente

Puede ser bastante sorprendente comprobar que no se solicita ningún tipo de identificación en la conexión a una compartición NFS. Se encontrará conectado sin tener que haber proporcionado credenciales. De hecho, NFS considera que los identificadores de los usuarios son coherentes entre el servidor y sus clientes, es decir, que todas las cuentas de usuario son idénticas en todas las máquinas y que sus identificadores de usuario (uid) son todos los mismos.

Cuando un cliente se conecta a una compartición NFS, presenta su uid y tiene exactamente los permisos que el usuario con el mismo uid en el servidor. No se realiza ningún control.

b. El caso particular del superusuario

Como la cuenta root tiene el uid 0 sea cual sea el sistema Linux, un cliente que se conecta al servidor con su cuenta de superusuario en teoría debería tener todos los privilegios en la compartición. Esta embarazosa situación se resuelve mediante la aplicación implícita de una opción de compartición: **root_squash**. En efecto, si un servidor recibe una solicitud de conexión de una cuenta con el uid 0, modifica su identificador y le aplica en la compartición el uid de una cuenta de servicio NFS. Esta cuenta (según la distribución nfsanonymous, nfsnobody, nobody...) en general sólo tendrá permisos del conjunto de usuarios "otros" en el sistema del servidor.

Compartición de datos con Samba

Samba es una solución software de interoperabilidad con Windows disponible en los sistemas Linux y Unix. El nombre de Samba viene del protocolo SMB (*Server Message Block*) utilizado para la compartición de recursos en las redes Microsoft. Permite en particular compartir archivos e impresoras en los servidores Linux para clientes Windows. La suite software Samba también tiene un cliente que permite a las máquinas Linux conectarse a los recursos compartidos de un servidor Windows.

1. Configuración general

a. Los daemons samba

Samba se basa en dos daemons llamados **nmbd** y **smbd**. El daemon **nmbd** se encarga de anunciar los servicios y en general de todo el funcionamiento NetBIOS over IP. El daemon **smbd** se encarga de las comparticiones de archivos y de impresoras.

El script de gestión del servicio que generalmente está presente en las distribuciones inicia estos dos daemons en cada arranque.

b. Los archivos de configuración

Los daemons samba tienen su configuración en el archivo de configuración **smb.conf**, generalmente ubicado en el directorio **/etc/samba**.

El archivo de configuración está dividido en secciones estandarizadas, cada una precedida por un título entre corchetes. Los parámetros de funcionamiento se ubican en cada una de estas secciones escritos siguiendo la sintaxis **parámetro = valor**.

Formato resumido de smb.conf

```
[sección1]
 parámetro1 = valor1
 parámetro2 = valor2
[sección2]
 parámetro3 = valor3
 parámetro4 = valor4
```

Existe una herramienta muy útil llamada **testparm** que valida el formato de un archivo de configuración samba. También devuelve un informe puro (sin líneas de comentarios) de la configuración por la salida estándar. Naturalmente, esta salida se puede redirigir a un archivo y generar un **smb.conf** legible y de tamaño razonable. Cabe destacar que el comando **testparm**ignora todo parámetro del archivo de configuración si se ha configurado con su valor por defecto. Este comportamiento se puede modificar con la opción **-v**. Entonces todas las opciones aplicables se mostrarán.

Ejemplo de uso de testparm para generar un archivo smb.conf sencillo

Este método se utiliza a menudo para usar un archivo de configuración con muchos comentarios obteniendo un archivo real de dimensiones razonables.

```
alfa:/etc/samba# mv smb.conf big.smb.conf
alfa:/etc/samba# wc -l big.smb.conf
326 big.smb.conf
alfa:/etc/samba# testparm big.smb.conf > smb.conf
Load smb config files from big.smb.conf
Processing section "[homes]"
Processing section "[printers]"
Processing section "[print$]"
Loaded services file OK.
Server role: ROLE_STANDALONE
Press enter to see a dump of your service definitions

alfa:/etc/samba# wc -l smb.conf
```

```
31 smb.conf
alfa:/etc/samba# testparm -v big.smb.conf > todas-las-opciones.info.smb.conf
alfa:/etc/samba#
```

*Las versiones preinstaladas de samba ofrecen siempre un archivo **smb.conf** preconfigurado. Aunque este archivo puede considerarse una buena base de partida, su tamaño (326 líneas para Debian) presenta el riesgo de impresionar a los iniciados. Seguramente sería mejor generar un archivo sólo con los elementos explícitamente necesarios.*

c. Configuración global

En su configuración más simple, una implementación samba incluye un servidor que alberga uno o más recursos. Algunos parámetros relacionados con el funcionamiento global y la identidad de este servidor se encuentran en una sección llamada **global** del archivo **smb.conf**.

En los ejemplos siguientes, nos pondremos en la situación de un servidor simple, fuera de un dominio Windows, que tiene comparticiones para clientes Windows.

Elementos comunes de la sección [global] en smb.conf

```
workgroup = grupo_de_trabajo
   server string = comentario
   log file = /ruta/log.%m
   max log size = log_maxi
   security = user (por defecto)
   encrypt passwords = true (por defecto)
```

Sección [global] del archivo smb.conf	
grupo_de_trabajo	El nombre del grupo de trabajo del servidor. Hay que tener en cuenta que este parámetro también proporciona el nombre del dominio cuando está trabajando en un dominio.
comentario	Comentario asociado al servidor. Visible por ejemplo en el Entorno de Red de las máquinas Windows.
log.%m	Definición del formato estándar de los archivos de registro.
log_maxi	Definición del tamaño máximo de los archivos de registro.
user	Opcional ya que es un parámetro por defecto. Parámetro de seguridad que obliga a autentificarse con una cuenta de usuario.
encrypt passwords	Opcional ya que es un parámetro por defecto. Necesario para todos los clientes modernos que presentan de forma nativa contraseñas encriptadas (desde NT4SP3).

2. Compartición de directorios

Para compartir un directorio hay que añadir una sección en el archivo **smb.conf**.

Formato típico de una sección de compartición en smb.conf

```
[nombre_compartición]
   comment = comentario
   path = ruta
   readonly = sólo_lectura
   browseable = yes
```

Declaración de comparticiones en smb.conf.	
nombre_compartición	El nombre con el que se verá la compartición en las máquinas Windows.
comentario	Opcional. Definición del comentario asociado a la compartición.
ruta	Definición de la ruta del directorio que se desea compartir. El

Si consulta el conjunto de

	directorio debe existir en el sistema de archivos Linux.
sólo_lectura	Definición del acceso a la compartición en modo de sólo lectura o lectura/escritura. *sólo_lectura* tendrá el valor yes o no según se elija la configuración. Hay que tener en cuenta que este parámetro se aplica a la compartición y que el acceso queda sometido a los permisos del sistema de archivos Linux.
browseable	Gestión de la visibilidad de la compartición para los clientes.

parámetros disponibles para el archivo **smb.conf**, puede quedar comprensiblemente impresionado por su gran cantidad. Hay que tener en cuenta que muchos parámetros funcionales pueden expresarse de varias formas. Tomemos por ejemplo el parámetro de acceso a una compartición en sólo lectura que hemos visto anteriormente. Todas las expresiones siguientes son equivalentes:

readonly = yes

readonly = true

writable = no

writable = false

writeable = no

writeable = false

3. Administración de credenciales

a. Algoritmos de hash y de almacenamiento de contraseñas

En la gran mayoría de sistemas operativos y aplicaciones las contraseñas no se almacenan sin encriptar. Las contraseñas de las cuentas están encriptadas y solamente se almacena la versión encriptada. La contraseña sin encriptar se olvida tan pronto como se encripta.

Cuando un usuario se conecta y teclea sus credenciales para identificarse, la contraseña se codifica de inmediato y esta versión recién encriptada de la contraseña se compara con la versión almacenada en la base de datos de cuentas de usuario del sistema. De este modo, la contraseña no se transmite nunca sin encriptar por la red.

Los algoritmos utilizados para encriptar la contraseña pertenecen a la familia de los algoritmos de hash. Funcionan de un modo un poco particular, en el sentido de que permiten encriptar pero nunca desencriptar datos: tienen un único sentido y por este hecho se los considera de una tipología diferenciada dentro del mundo de la criptografía. Este modo de funcionamiento justifica por qué, cuando un usuario pierde su contraseña, se le puede reasignar una nueva, pero no se le puede decir cuál era la que ha olvidado. La única información que se guarda es la versión encriptada de la contraseña y es hipotéticamente indescifrable.

Los algoritmos de hash más comunes se llaman MD4, MD5 y SHA1. Se utilizan para almacenar contraseñas, las operaciones de firma digital y los controles de integridad.

b. Autentificación con servidores Samba

Un servidor Linux con la suite software Samba instalada utiliza nativamente las cuentas del sistema para las autentificaciones Samba. De este modo, toda conexión por parte de un cliente se realiza con una cuenta de usuario albergada en el sistema Linux. Sin embargo, esta situación podría ser un problema. El cliente de Windows presentará una contraseña encriptada por el algoritmo de hash nativo de los sistemas Windows MD4 (*Message Digest 4*), mientras que las contraseñas de los sistemas Linux usan el algoritmo MD5 (*Message Digest 5*). La contraseña encriptada proporcionada por el cliente Windows no será, por tanto, la misma que la que está almacenada en el archivo **/etc/shadow** del sistema Linux. Por consiguiente, la autentificación será imposible, aunque la contraseña sin cifrar sea la misma.

Para que los clientes Windows se puedan autentificar en los sistemas Linux, hay que hacer que estos sistemas alberguen una versión de la contraseña encriptada en MD4 además de la contraseña nativa encriptada en MD5. Estas dos contraseñas se administrarán de forma independiente y podrán ser incluso distintas.

c. Generación de contraseñas MD4

El comando específico **smbpasswd** permite crear una contraseña MD4 para una cuenta Linux existente. Esta contraseña se almacenará aparte, generalmente en el archivo **/etc/samba/smbpasswd**.

Sintaxis del comando smbpasswd para asignar una contraseña

```
smbpasswd -a nombre_cuenta
```

Comando smbpasswd: opciones y parámetros	
-a	Opcional. Necesario si la cuenta no dispone todavía de una contraseña samba.
nombre_cuenta	La cuenta Linux a la que hay que asignar la contraseña samba.

d. Sincronización con contraseñas Linux

Se puede solicitar que se sincronicen las contraseñas samba con las contraseñas del sistema Linux. Atención, tal y como se ha explicado anteriormente, las contraseñas se encriptan en ambos sistemas con algortimos de hash distintos, que son por definición irreversibles. La sincronización sólo se puede realizar en el momento en que la contraseña se introduce sin encriptar cuando se utiliza el comando **smbpasswd**. Es entonces cuando la contraseña se encripta dos veces con los dos algoritmos diferentes y se modifican las dos bases de datos de cuentas de usuario. Esta sincronización se activa mediante una directiva en el archivo **smb.conf**.

Activación de la sincronización de contraseñas en smb.conf

```
unix password sync = yes
```

e. Borrado o desactivación de una cuenta samba

Se puede necesitar interrumpir el acceso de una cuenta de un usuario a los recursos compartidos del servidor samba. El comando **smbpasswd** puede eliminar, desactivar o reactivar la cuenta samba, independientemente de la cuenta Linux asociada.

Comando smbpasswd para desactivar una cuenta samba

```
smbpasswd -d nombre_cuenta
```

Comando smbpasswd para reactivar una cuenta samba

```
smbpasswd -e nombre_cuenta
```

Comando smbpasswd para eliminar una cuenta samba

```
smbpasswd -x nombre_cuenta
```

Donde *nombre_cuenta* representa la cuenta de usuario samba que se desea modificar. Cabe decir que las operaciones en las cuentas samba no tienen efecto alguno en la cuenta Linux correspondiente.

4. El cliente Samba

El cliente Samba permite acceder a una compartición de una máquina Windows o Samba desde un cliente Linux. Permite incluso a un cliente Linux conectarse a un servidor Samba Linux, pero el objetivo es más bien acceder a datos de una compartición Windows desde una máquina Linux. Los dos comandos principales del cliente samba son **smbclient** y **smbmount**.

a. Uso puntual de recursos compartidos con smbclient

Básicamente, se utiliza **smbclient** para obtener información de los recursos compartidos albergados por un servidor SMB.

Utilización de smbclient para obtener información acerca de un servidor smb

```
smbclient -L dirección_servidor -U nombre_usuario
```

smbclient para mostrar comparticiones: parámetros	
dirección_servidor	La dirección IP del servidor del que se quiere mostrar los recursos.
nombre_usuario	Indica el nombre del usuario que realiza la consulta al servidor. Tiene que ser una cuenta existente y válida en el servidor.

También se puede utilizar el comando **smbclient** de forma interactiva conectándose a un recurso compartido y accediendo a un shell que permita realizar operaciones con los archivos.

Utilizaciones de smbclient en modo interactivo

```
smbclient \\\\dirección_servidor\\compartición -U nombre_usuario
smbclient //dirección_servidor/compartición -U nombre_usuario
```

Donde compartición representa el nombre de la compartición albergada por el servidor. Las múltiples contrabarras son necesarias aunque generen una sintaxis un tanto curiosa. De hecho, se trata de una ruta UNC (*Uniform Naming Convention*), utilizada para designar un recurso en los entornos Windows. Una ruta UNC se compone del nombre del servidor, precedido de dos contrabarras, seguido de la ruta al recurso, separando con una contrabarra cada nivel. Sin embargo, se da el caso que en los sistemas Linux la contrabarra es un carácter reservado que indica que el shell no debe interpretar el carácter siguiente. Para escribir una contrabarra de verdad, hay que anteponerle otra para indicarle al sistema que la segunda debe considerarse como una contrabarra normal. Una alternativa más ligera consiste en cambiar las contrabarras por barras normales. Ambas sintaxis están permitidas.

Una vez que este comando se ejecuta y después de haber introducido la contraseña del usuario, se entra en el shell específico **smbclient** que permite realizar operaciones con los archivos. Los principales usos serán por supuesto obtener o enviar archivos a la compartición. Se puede desplazar por la estructura de directorios con el comando **cd**. Además, los dos comandos básicos son **get** para obtener archivos y **put** para enviar archivos a la compartición.

Ejemplo de uso de smbclient en modo interactivo

La utilidad smbclient presenta un conjunto de comandos parecido al de los clientes FTP.

```
alfa:~# smbclient \\\\192.168.0.1\\data -U toto
Enter toto's password:
Domain=[WSERVIDOR] OS=[Windows 5.1] Server=[Windows 2000 LAN Manager]
smb: \> ls
  .                                   D        0  Wed Feb  3 19:28:33 2010
  ..                                  D        0  Wed Feb  3 19:28:33 2010
  dos                                 D        0  Wed Feb  3 18:50:05 2010
  uno                                 D        0  Wed Feb  3 19:28:38 2010

    40915 blocks of size 262144. 34718 blocks available
smb: \> cd uno
smb: \uno\> ls
  .                                   D        0  Wed Feb  3 19:28:38 2010
  ..                                  D        0  Wed Feb  3 19:28:38 2010
  archivo.txt                         A       27  Wed Feb  3 19:15:49 2010
  truco.bmp                           A        0  Wed Feb  3 18:46:44 2010

    40915 blocks of size 262144. 34718 blocks available
smb: \uno\> get archivo.txt
getting file \uno\archivo.txt of size 27 as archivo.txt (2,0 kb/s) (average 2,0 kb/s)
smb: \uno\> exit
alfa:~# ls
archivo.txt
alfa:~#
```

b. Montaje de una compartición smb con smbmount

Aunque **smbclient** permite realizar un acceso puntual a las compartición es, existe un método más cómodo para usar directorios compartidos desde un cliente Linux: el montaje de una compartición en la estación de trabajo Linux.

El comando **smbmount** permite realizar el montaje de una compartición SMB en un directorio local tal y como se puede hacer con un sistema de archivos local o una compartición NFS.

Sintaxis del comando smbmount

```
smbmount \\\\dirección_servidor\\compartición punto_de_montaje -o user=nombre_usuario
smbmount //dirección_servidor/compartición punto_de_montaje -o user=nombre_usuario
```

smbmount: opciones y parámetros	
dirección_servidor	Dirección IP del servidor que tiene la compartición a la que se quiere acceder.
compartición	Nombre de la compartición albergada en el servidor.
punto_de_montaje	Directorio existente en el que se montará la compartición.
nombre_usuario	Nombre del usuario que realizará la petición al servidor. Tiene que ser una cuenta existente y válida en el servidor.

Existe una alternativa a esta sintaxis, que supone realizar el montaje mediante el comando **mount**llamando a **smbmount** como subprograma. Esta sintaxis presenta la ventaja de uniformar todas las operaciones de montaje y, por lo tanto, sólo tener que recordar una sintaxis genérica.

Sintaxis del comando mount para montar una compartición smb

```
mount -t smbfs -o username=nombre_usuario //dirección_servidor/compartición
punto_de_montaje
```

La opción **-t smbfs** provoca la llamada al subprograma **smbmount** para realizar el montaje, pero a partir de una sintaxis casi estándar para realizar el montaje.

c. Montaje de una compartición CIFS

Para responder a las necesidades de apertura del protocolo, SMB se ha normalizado, ha evolucionado y se ahora se denomina CIFS (*Common Internet File System*). La suite software Samba ahora designa a su cliente y a los elementos software con este nombre. Como los hábitos resisten a morir, todavía persiste el uso del nombre de SMB.

Según las versiones de Samba usadas, se puede utilizar smb, cifs o smb y cifs indiferentemente. La tendencia es la desaparición de smb en beneficio de cifs.

Sintaxis del comando mount para compartición es cifs

```
mount -t cifs -o username=nombre_usuario //dirección_servidor/compartición
punto_de_montaje
```

Se puede verificar desde el lado servidor cuáles son los clientes que están conectados. El comando **smbstatus** muestra las conexiones smb activas.

Compartición de archivos con FTP

1. El protocolo FTP

a. Historia

FTP (*File Transfer Protocol*) es un protocolo cliente/servidor bastante antiguo que fue uno de los primeros que permitía compartir archivos entre ordenadores. Tiene un pasado glorioso y, por ejemplo, ya se usaba antes de la creación del protocolo SMTP para transmitir mensajes electrónicos de un ordenador a otro.

Hoy en día, su edad y una cierta rigidez lo hacen menos apto para compartir archivos cómodamente. Sin embargo, se sigue usando con asiduidad, especialmente por los servicios de alojamiento web que ofrecen a sus clientes accesos FTP para actualizar sus páginas web.

b. Parámetros técnicos

El nivel de transporte de FTP es TCP y funciona por el puerto 21 para la transmisión de comandos. El puerto 20 es el que se usa tradicionalmente para transmitir datos, pero no es un comportamiento universal.

FTP soporta la autentificación de clientes, pero con un grado de seguridad débil que lo convierte en un protocolo inadecuado para la transmisión de archivos sensibles. En efecto, FTP es conocido por transportar la contraseña de sus clientes sin encriptar. Por estas razones, hoy en día FTP tiene un uso específico: el modo anónimo. Los servidores FTP pueden reconocer una cuenta única anónima y autorizarle un acceso limitado, generalmente en modo sólo lectura en algunos directorios. La cuenta tiene que llamarse obligatoriamente **anonymous**, y el servidor puede solicitar una contraseña, sin importar el juego de caracteres. La contraseña se guardará por motivos de trazabilidad aunque el cliente no tenga la obligación de introducir una contraseña.

c. Modo FTP activo y FTP pasivo

Tradicionalmente, los clientes FTP trabajaban en **modo activo**, donde la sesión se establecía en el puerto 21 del servidor y los datos se enviaban por iniciativa del servidor desde el puerto 20 a un puerto cualquiera del cliente. Este modo de funcionamiento, que existe desde antes de la generalización de los cortafuegos, no está exento de problemas, ya que es visto por el cortafuegos como una sesión abierta desde el servidor a un puerto impredecible del cliente.

El **modo pasivo** apareció para corregir esta situación mediante el establecimiento de dos sesiones por el cliente. El puerto utilizado para los datos es uno cualquiera que lo anuncia el servidor en modo comando y lo utiliza el cliente para abrir la sesión de datos.

2. Los clientes FTP

a. Los clientes FTP gráficos

Son muchos los clientes FTP gráficos existentes y están disponibles para todas las plataformas. Se puede citar filezilla, que es un producto open source muy popular en los sistemas Windows. La configuración y el uso de clientes FTP gráficos varían según el producto y no presentan gran dificultad, su uso no se tratará en este libro.

b. El cliente FTP por línea de comandos

La mayoría de los sistemas incluyen un cliente FTP por línea de comandos. El modo de funcionamiento de estos clientes puede hacerlos más incómodos para un uso frecuente pero son extremadamente prácticos para comprobar la configuración de un servidor FTP.

La carga de estos clientes se realiza de la forma más sencilla mediante el comando **ftp**.

La ventaja principal del cliente FTP por línea de comandos es que permite realizar todas las operaciones deseadas una a una y, por lo tanto, en caso de error ver dónde está el error. Por el contrario, los clientes

gráficos tienen tendencia a automatizar un gran número de operaciones. Para la conexión FTP con Internet Explorer, por ejemplo, la conexión es anónima y se envía una contraseña estándar automáticamente.

Cliente FTP: comandos comunes	
open	Abre una sesión FTP con el servidor al que se ha hará referencia. El cliente solicitará de manera interactiva la dirección del servidor.
close	Cierra la sesión FTP en curso.
ls	Muestra los archivos contenidos en el directorio remoto actual.
cd	Cambia el directorio remoto actual. La sintaxis es la misma que la de un shell Linux.
get	Descarga un archivo del directorio remoto actual en el directorio local actual.
put	Sube (envía) un archivo del directorio local actual al directorio remoto actual.

3. El servidor Pure-FTPd

Pure-FTPd es un servidor FTP cuyo objetivo es ofrecer un servicio de transferencia de archivos simple, estable y eficaz. Su objetivo es ser apto tanto para principiantes como para entornos de producción en el lugar de trabajo. Su característica principal es que puede ser iniciado fácilmente por línea de comandos sin necesitar un archivo de configuración.

a. Funcionamiento para accesos de usuarios a sus directorios personales

Este es el funcionamiento por defecto. Los usuarios dispondrán de una cuenta de usuario y un directorio personal donde podrán acceder con su identificador y su contraseña habituales. Atención: generalmente se desaconseja este modo de funcionamiento ya que la contraseña, circulando sin encriptar, supone un peligro para las contraseñas Linux de los usuarios.

Inicio del servicio

```
pure-ftpd
```

b. Funcionamiento para accesos anónimos

Se puede activar el acceso anónimo si se ha creado una cuenta de usuario llamada **ftp** en el servidor. Los clientes conectados en modo anónimo trabajan entonces en el directorio **/home/ftp**.

Se puede trabajar en modo anónimo llamando a pure-ftpd con la opción **--anonymousonly**.

c. Opciones de funcionamiento

Pure-ftpd funciona generalmente sin archivo de configuración. A la línea de comandos que inicia el servicio se le pueden añadir opciones de configuración en función del resultado deseado. Sin embargo, algunas implementaciones utilizan uno o varios archivos de configuración que el script de inicio del servicio interpretará. La lista mostrada a continuación presenta algunas de las opciones más comunes.

pure-ftpd: opciones comunes	
--help	Muestra las opciones posibles.
--displaydotfile	También muestra los archivos escondidos a los clientes.
--anonymousonly	Funcionamiento del servidor sólo en modo anónimo (si la cuenta ftp existe).
--noanonymous	Impide cualquier intento de conexión anónimo (incluso si la cuenta ftp existe).
--maxidletime	Tiempo máximo de inactividad antes de forzar la desconexión.
--anonymouscantupload	Impide a los usuarios anónimos transferir archivos al

	servidor.
--anonymouscancreatedirs	Permite a los usuarios anónimos crear directorios.

4. El servidor vsftpd

vsftpd (*very secure FTP daemon*) es otro servidor FTP muy popular en los sistemas Linux. Necesita un servicio y un archivo de configuración: **vsftpd.conf**. La certificación LPI requiere un conocimiento mínimo de vsftpd.

Formato de las opciones para el archivo vsftpd.conf

```
parámetro=valor
```

La mayoría de los parámetros tienen como posibles valores YES y NO.

Archivo vsftpd.conf: parámetros comunes	
anonymous_enable	Autoriza o no el acceso anónimo.
local_enable	Autoriza o no a los usuarios a acceder a su directorio personal.
write_enable	Autoriza o no la subida de archivos al servidor.
anon_upload_enable	Autoriza o no la subida de archivos al servidor a usuarios anónimos.
anon_mkdir_write_enable	Autoriza o no la creación de directorios a usuarios anónimos.

Comprobación de los conocimientos adquiridos: preguntas/respuestas

Ponga a prueba sus conocimientos respondiendo a las preguntas siguientes. Estas preguntas no siempre esperan una respuesta cerrada. Las preguntas planteadas en la certificación, a pesar de que abordan los mismos temas, en su mayoría serán planteadas en forma de preguntas de opción múltiple o que requieran una respuesta corta, en palabras escritas en el teclado.

1. Preguntas

1. ¿Cómo acceden los clientes de línea de comandos a los datos de una compartición, ya sea del tipo NFS o Samba?
2. ¿Se pueden compartir los directorios a los que hace referencia el archivo /etc/exports sin haber iniciado el servicio NFS?
3. ¿Por qué la opción root_squash se aplica por defecto a una compartición NFS?
4. ¿Qué proceso inicia la carga de un script de gestión del servicio NFS?
5. En una conexión a un servidor NFS no fiable, ¿qué opción sería la adecuada para asegurar a un cliente NFS que las operaciones de escritura se realizan formalmente?
6. ¿Se puede comprobar la validez de un archivo de configuración Samba sin cargar el servicio?
7. ¿Cómo se impide a los usuarios que vean una compartición Samba en el Entorno de Red?
8. ¿Cómo se crea una contraseña a partir de la contraseña unix de una cuenta previamente existente en el sistema?
9. ¿Se puede sincronizar las contraseñas Unix con las contraseñas Samba?
10. ¿Por qué el modo activo ha ido cayendo en desuso en beneficio del modo pasivo en los clientes FTP?

2. Respuestas

1. ¿Cómo acceden los clientes de línea de comandos a los datos de una compartición, ya sea del tipo NFS o Samba?

   Mediante una operación de montaje. El directorio compartido se monta en un directorio local. Cuidado: aunque se use el comando universal mount, los elementos software cliente deben haberse instalado en el sistema para que la operación de montaje tenga éxito.

2. ¿Se pueden compartir los directorios a los que hace referencia el archivo /etc/exports sin haber iniciado el servicio NFS?

   Sí, con el comando exportfs, llamado con el parámetro -a.

3. ¿Por qué la opción root_squash se aplica por defecto a una compartición NFS?

   Porque el control de acceso a las comparticiones NFS se basa en el identificador (uid) de los clientes que se conectan. La cuenta root tiene siempre el identificador 0 y la opción root_squash hace que sus privilegios desaparezcan de manera que el usuario root no tenga plenos derechos en una compartición NFS. Sin embargo, este comportamiento se puede modificar con la opción no_root_squash en la carga de la compartición.

4. ¿Qué proceso inicia la carga de un script de gestión del servicio NFS?

   NFS, de hecho, depende de tres procesos: portmapd, nfsd y mountd. El nombre del script de inicio del servicio varía según la distribución (nfs para Red Hat, nfk-kernel-server para Debian).

5. En una conexión a un servidor NFS no fiable, ¿qué opción sería la adecuada para asegurar a un cliente NFS que las operaciones de escritura se realizan formalmente?

   La opción de montaje sync impide que se produzcan escrituras asíncronas. De este modo, el servidor prohíbe que se use la caché en escritura. El cliente no será informado del éxito de una operación de escritura hasta que ésta se haya producido físicamente.

6. ¿Se puede comprobar la validez de un archivo de configuración Samba sin cargar el servicio?

   Sí, con el comando testparm. El comando testparm comprueba la validez del archivo de configuración y muestra las comparticiones activas por la salida estándar. Cabe decir que los comandos por defecto no se

muestran a no ser que se invoque con la opción -v.

7 ¿Cómo se impide a los usuarios que vean una compartición Samba en el Entorno de Red?

El parámetro browseable en la definición de una compartición en el archivo de configuración permite gestionar la visibilidad de una compartición para los clientes.

8 ¿Cómo se crea una contraseña a partir de la contraseña Unix de una cuenta previamente existente en el sistema?

No se puede. La contraseña de una cuenta Unix se encripta con el algoritmo de hash MD5, irreversible por definición. Las contraseñas SMB tienen que encriptarse con el algoritmo MD4, no se puede crear esta contraseña a partir de los datos encriptados presentes en el sistema.

9 ¿Se puede sincronizar las contraseñas Unix con las contraseñas Samba?

Sí, incluyendo la directiva "unix passwd sync = yes" en el archivo de configuración. Atención: los algoritmos de hash son distintos, esta sincronización sólo se podrá realizar cuando la contraseña Samba se exprese sin encriptar. Entonces se producen dos operaciones de encriptación: una en MD5 para la base de datos de cuentas Unix y otra en MD4 para la base de datos de cuentas Samba.

10 ¿Por qué el modo activo ha ido cayendo en desuso en beneficio del modo pasivo en los clientes FTP?

Porque el modo activo utilizado tradicionalmente usa un número de puerto para los datos iniciado por el servidor hacia el cliente y generalmente los cortafuegos ven esta acción como una amenaza. Con el modo pasivo, el cliente inicia tanto las sesiones de comandos como las de datos, evitando que salten las alarmas de los cortafuegos.

Trabajos prácticos

1. Despliegue de comparticiones Samba en el servidor alfa

Hay una gran cantidad de estaciones de trabajo Windows en su red y desea proporcionarles un servidor de archivos. Por lo tanto, decide instalar el servicio Samba en el servidor alfa. Este servidor debe permitir a los usuarios acceder a los documentos de su directorio personal en el servidor y también publicar una compartición común de tipo "cajón de sastre" para el intercambio libre de datos entre usuarios.

a. Instalación de los servicios software

En el servidor alfa, instale el middleware Samba mediante el comando siguiente:

```
apt-get install samba
```

Acepte las opciones por defecto.

b. Visualización de la configuración por defecto

Comandos útiles

- vi
- testparm

Operaciones

1. Con el servicio recién instalado, mostrar los parámetros activos aplicados por el servidor provenientes del archivo smb.conf.
2. Mostrar ahora los parámetros activos, pero esta vez incluyendo los parámetros por defecto no mencionados de forma explícita en el archivo smb.conf.

Resumen de los comandos y resultado por pantalla

Parámetros explícitos:

```
alfa:/etc/samba# testparm
Load smb config files from /etc/samba/smb.conf
Processing section "[homes]"
Processing section "[printers]"
Processing section "[print$]"
Loaded services file OK.
Server role: ROLE_STANDALONE
Press enter to see a dump of your service definitions

[global]
        server string = %h server
        obey pam restrictions = Yes
        passdb backend = tdbsam
        pam password change = Yes
        passwd program = /usr/bin/passwd %u
        passwd chat = *Enter\snew\s*\spassword:* %n\n *Retype\snew\s*\spassword:* %n\n
*password\supdated\ssuccessfully* .
        unix password sync = Yes
        syslog = 0
        log file = /var/log/samba/log.%m
        max log size = 1000
        dns proxy = No
        panic action = /usr/share/samba/panic-action %d

[homes]
        comment = Home Directories
        valid users = %S
```

Todos los

```
        create mask = 0700
        directory mask = 0700
        browseable = No
(...)
alfa:/etc/samba#
```

parámetros:

```
alfa:/etc/samba# testparm -v
Load smb config files from /etc/samba/smb.conf
Processing section "[homes]"
Processing section "[printers]"
Processing section "[print$]"
Loaded services file OK.
Server role: ROLE_STANDALONE
Press enter to see a dump of your service definitions

[global]
        dos charset = CP850
        unix charset = UTF-8
        display charset = LOCALE
        workgroup = WORKGROUP
        realm =
        netbios name = ALPHA
        netbios aliases =
        netbios scope =
        server string = %h server
        interfaces =
        bind interfaces only = No
        config backend = file
        security = USER
        auth methods =
        encrypt passwords = Yes
        update encrypted = No
        client schannel = Auto
(... ¡375 líneas en total!)
alfa:/etc/samba#
```

c. Gestión de las contraseñas

Comandos útiles

- smbpasswd

Operaciones

1. Asignar una contraseña Samba a la cuenta del usuario usuario presente en el servidor alfa.

Resumen de los comandos y resultado por pantalla

```
alfa:/etc/samba# smbpasswd -a usuario
New SMB password:
Retype new SMB password:
alfa:/etc/samba#
```

d. Acceso de los usuarios a sus directorios personales desde la estación de trabajo

Todos los usuarios que disponen de una máquina Windows están ausentes. Usted decide realizar las primeras pruebas funcionales desde la estación de trabajo Ubuntu. Para ello, utilizará el cliente gráfico de la estación de trabajo.

Comandos útiles

- Utilización de la interfaz gráfica

Operaciones

1. Abra una sesión en la estación de trabajo.
2. En el menú **Lugares**, haga clic en **Conectar con el servidor**.
3. En el desplegable **Tipo de servicio**, elija **Compartido por Windows**.
4. Rellene el campo **Servidor** con la dirección IP de alfa.
5. Rellene el campo **Compartido** con el nombre **usuario**.
6. Haga clic en el botón **Conectar**.
7. En la ventana de autentificación, introduzca el password del usuario.
8. El directorio del usuario tiene que estar ahora accesible.

e. Creación de una compartición común

Comandos y archivos útiles

- chmod
- mkdir
- smb.conf
- testparm
- vi

Operaciones

1. En alfa, crear el directorio **/public**.
2. Hacer que todos los usuarios puedan leer y escribir en este directorio.
3. Editar el archivo de configuración Samba en alfa.
4. Añadir una sección de compartición accesible en modo lectura/escritura para el directorio public.
5. Hacer que esta compartición sea visible cuando se navegue por la red (del tipo Entorno de Red de Windows).
6. Hacer que el contenido de los directorios se pueda borrar por todos (permisos **rwx**para los usuarios **other** en las carpetas creadas).
7. Comprobar la validez de la sintaxis sin reiniciar el servicio.
8. Reiniciar el servicio samba.

Resumen de los comandos y resultado por pantalla

Creación del directorio:

```
alfa:/home/usuario# mkdir /public
alfa:/home/usuario# chmod o+rwx /public/
alfa:/home/usuario#
```

Sección de la compartición en smb.conf:

Comprobación de la sintaxis:

```
[public]
  path = /public
  writeable = yes
  browseable = yes
  directory mask = 0777
```

Reinicio del servicio:

```
alfa:/public# testparm
Load smb config files from /etc/samba/smb.conf
Processing section "[homes]"
Processing section "[public]"
Processing section "[printers]"
Processing section "[print$]"
Loaded services file OK.
Server role: ROLE_STANDALONE
Press enter to see a dump of your service definitions
```

f. Acceso de los usuarios a la nueva compartición

```
alfa:/home/usuario# /etc/init.d/samba reload
alfa:/home/usuario#
```

Comandos útiles

- Utilización de la interfaz gráfica

Operaciones

1. Abrir una sesión en la estación de trabajo.
2. En el menú **Lugares**, haga clic en **Conectar con el servidor**.
3. En el desplegable **Tipo de servicio**, elija **Compartido por Windows**.
4. Rellene el campo **Servidor** con la dirección IP de alfa.
5. No rellene el campo **Compartido** para visualizar todas las comparticiones configuradas como visibles.
6. Haga clic en el botón **Conectar**.
7. En la ventana de autentificación, introduzca la contraseña del usuario.
8. El directorio public tiene que estar ahora visible. Observe que el directorio personal no aparece, ya que se ha configurado como no visible (Browseable = No).

Despliegue de comparticiones NFS en el servidor beta

La próxima instalación de una solución de virtualización necesita el despliegue de un servidor NFS en la red. El acceso se realizará con la cuenta root y se deberá permitir la escritura en la compartición. Usted decide configurar un servicio NFS en el servidor beta.

a. Instalación de los servicios software

En la estación de trabajo Ubuntu, instale el middleware NFS mediante el comando siguiente:

```
sudo apt-get install nfs-common
```

Los servicios NFS ya deberían estar disponibles en el servidor beta.

b. Configuración de la compartición

Comandos y archivos útiles

- /etc/exports
- exportfs
- mkdir
- vi

Operaciones

1. Arrancar el servicio NFS del servidor beta.
2. Comprobar que actualmente no hay ninguna compartición activa.
3. Crear el directorio **/virtu**.
4. Crear el archivo de configuración **/etc/exports** que compartirá este directorio en modo lectura/escritura con acceso normal en la cuenta de root.
5. Desplegar la compartición sin reiniciar el servicio nfs. Comprobar.

Resumen de los comandos y resultado por pantalla

Inicio del servicio:

```
[root@beta init.d]# service nfs start
Inicio de los servicios NFS:
[  OK  ]
Iniciando cuotas NFS:
[  OK  ]
Inicialización del demonio NFS:
[  OK  ]
Inicialización de NFS mountd:
[  OK  ]
[root@beta init.d]#
```

Comprobación de las comparticiones activas:

```
[root@beta init.d]# exportfs
[root@beta init.d]#
```

Creación del directorio:

Archivo

```
[root@beta init.d]# mkdir /virtu
[root@beta init.d]#
```

/etc/exports:

```
/virtu *(rw,no_root_squash)
```

Despliegue de la compartición:

c.

```
[root@beta init.d]# exportfs -a
[root@beta init.d]# exportfs
/virtu          <world>
```

```
[root@beta init.d]#
```

Conexión desde la estación de trabajo cliente

Comandos útiles

- mkdir
- mount

Operaciones

1. Crear el directorio **virtu** en **/mnt** que servirá de punto de montaje.
2. Montar la compartición NFS **/virtu** del servidor beta en el punto de montaje**/mnt/virtu/**.

Resumen de los comandos y resultado por pantalla

Creación del punto de montaje:

```
usuario@estacion:/mnt$ sudo mkdir /mnt/virtu
[sudo] password for toto:
usuario@estacion:/mnt$ ls
virtu
usuario@estacion:/mnt$
```

Montaje de la compartición:

```
usuario@estacion:/mnt$ sudo mount -t nfs 192.168.200.102:/virtu virtu
usuario@estacion:/mnt$ ls virtu
dos  tres  uno
usuario@estacion:/mnt$
```

3.

Configuración de un servidor FTP en el servidor alfa

Se da un caso muy excepcional en el que algunos usuarios tienen que enviar archivos muy voluminosos por correo electrónico. Usted decide entonces desplegar un servidor FTP. Un poco inquieto en materia de seguridad, decide que el servicio se cargará bajo demanda y permitirá a usuarios anónimos enviar y descargar archivos, pero sin que puedan consultar el contenido del directorio de trabajo FTP.

a. Instalación del servicio software

En el servidor alfa, instale la aplicación pure-ftpd mediante el comando siguiente:

```
apt-get install pure-ftpd
```

b. Configuración e inicio del servicio

Comandos útiles

- adduser
- chmod
- passwd
- pure-ftpd

Operaciones

1. Añadir una cuenta de usuario ftp.
2. Bloquear las cuentas de usuario y limitar los permisos en su directorio personal. Se debe poder escribir y crear documentos, pero no verlos. El grupo y otros usuarios no deben tener permisos en el directorio.
3. Iniciar el servicio funcionando sólo en modo anónimo.

Resumen de los comandos y resultado por pantalla

Creación de la cuenta ftp:

```
alfa:/# adduser ftp
Añadiendo el usuario `ftp' ...
Añadiendo el nuevo grupo `ftp' (1001) ...
Añadiendo el nuevo usuario `ftp' (1001) con grupo `ftp' ...
Creando del directorio personal `/home/ftp' ...
Copiando los ficheros desde `/etc/skel' ...
Introduzca la nueva contraseña UNIX:
Vuelva a escribir la nueva contraseña UNIX:
passwd: contraseña actualizada correctamente
Cambiando la información de usuario para ftp
Introduzca el nuevo valor, o presiones ENTER para el predeterminado
        Nombre completo []: ftp user
        Número de habitación []:
        Teléfono del trabajo []:
        Teléfono de casa []:
        Otro []:
¿Es correcta la información? [S/n]
alfa:/#
```

Bloqueo de la cuenta:

```
alfa:/# passwd -l ftp
Contraseña cambiada.
alfa:/#
```

Limitación de permisos:

```
alfa:/# chmod u=wx,go= /home/ftp
alfa:/# ls -ld /home/ftp
d-wx------ 2 ftp ftp 4096 jul 19 00:43 /home/ftp
alfa:/#
```

Inicio del servicio:

```
alfa:/#pure-ftpd --anonymous_only
```

c.

Conexión desde la estación de trabajo cliente Ubuntu

Comandos útiles

- ftp
- vi

Operaciones

1. Crear un archivo de texto con el método que se desee.
2. Iniciar el cliente FTP.
3. Abrir una sesión FTP anónima con el servidor alfa.
4. Intentar ver el contenido del directorio.
5. Enviar su archivo al servidor.

Resumen de los comandos y resultado por pantalla

Creación del archivo:

```
usuario@estacion:~$ echo "bla bla" > archivo
usuario@estacion:~$ echo "bla" >> archivo
usuario@estacion:~$ cat archivo
bla bla
bla
usuario@estacion:~$
```

Apertura de la sesión FTP:

```
usuario@estacion:~$ ftp
ftp> open 192.168.200.101
Connected to 192.168.200.101.
220---------- Welcome to Pure-FTPd [privsep] [TLS] ----------
220-You are user number 1 of 50 allowed.
220-Local time is now 14:12. Server port: 21.
220-Only anonymous FTP is allowed here
220-IPv6 connections are also welcome on this server.
220 You will be disconnected after 15 minutes of inactivity.
Name (192.168.200.101:toto): anonymous
230 Anonymous user logged in
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

Prueba de lectura del contenido del directorio:

```
ftp> ls
200 PORT command successful
150 Connecting to port 49524
226-Sorry, we were unable to read [.]
226-Options: -l
226 0 matches total
ftp>
```

Envío del archivo al servidor:

```
ftp> put archivo
local: archivo remote: archivo
200 PORT command successful
150 Connecting to port 50945
226-File successfully transferred
226 0.006 seconds (measured here), 1.93 Kbytes per second
12 bytes sent in 0.00 secs (5.8 kB/s)
ftp>
ftp> bye
221-Goodbye. You uploaded 1 and downloaded 0 kbytes.
```

```
221 Logout.
usuario@estacion:~$
```

Requisitos y objetivos

1. Requisitos

Los conocimientos adquiridos en la certificación LPI nivel 1, especialmente:

- ☒ Saber editar archivos de texto.
- ☒ Tener conocimientos generales de redes TCP/IP.

2. Objetivos

Al final de este capítulo, será capaz de:

- ☒ Conocer la arquitectura y el principio de la resolución DNS.
- ☒ Conocer los principales tipos de registro DNS.
- ☒ Configurar un cliente DNS.
- ☒ Configurar un servidor de caché DNS.
- ☒ Configurar una redirección de la resolución DNS.
- ☒ Usar el comando de control rndc.
- ☒ Administrar zonas DNS directas e inversas.
- ☒ Crear registros de recursos en zonas DNS.
- ☒ Administrar zonas DNS secundarias.
- ☒ Configurar una delegación de zona DNS.
- ☒ Conocer las principales utilidades de comprobación de resolución DNS.
- ☒ Asegurar un servidor DNS.

Características generales

El sistema DNS es el soporte de un gran abanico de funcionalidades de Internet, que abarca desde la navegación hasta el envío de correos electrónicos. Es esencial configurarlo correctamente en el ámbito de las redes locales, y es primordial en Internet.

1. Los inicios de la resolución de nombres y la aparición de DNS

Desde el comienzo de las redes IP, el objetivo principal de la resolución de nombres es hacer corresponder un nombre fácil de recordar con una dirección IP, el único dato realmente útil para contactar con una máquina remota.

nombre-de-la-máquina <--> 130.130.28.12

Mientras la cantidad de máquinas públicas en Internet era pequeña, todas las resoluciones se realizaban mediante un archivo llamado **hosts** que se descargaba a intervalos regulares para tener actualizadas las últimas novedades.

El DNS se creó como solución a los límites del archivo **hosts** descargado y tenía que cumplir con ciertos requisitos de diseño.

El DNS es dinámico

Los registros se tienen que poder añadir de una forma única al sistema y estar rápidamente disponible para todos.

El DNS se replica

No se puede permitir depender de un solo servidor. Hay que disponer de la información existente siempre en varias copias.

El DNS está jerarquizado

Los datos se clasifican en una estructura de árbol que permite su organización. Cada nivel de la jerarquía se llama "zona" y la cima de esta jerarquía es la zona ".".

El DNS es un sistema distribuido

Los datos se reparten en una multitud de "subbases de datos" (las zonas DNS) y el conjunto de estas pequeñas bases de datos compone la totalidad de los registros DNS. Este funcionamiento tiene como ventaja la fácil administración repartiendo la carga en miles de servidores.

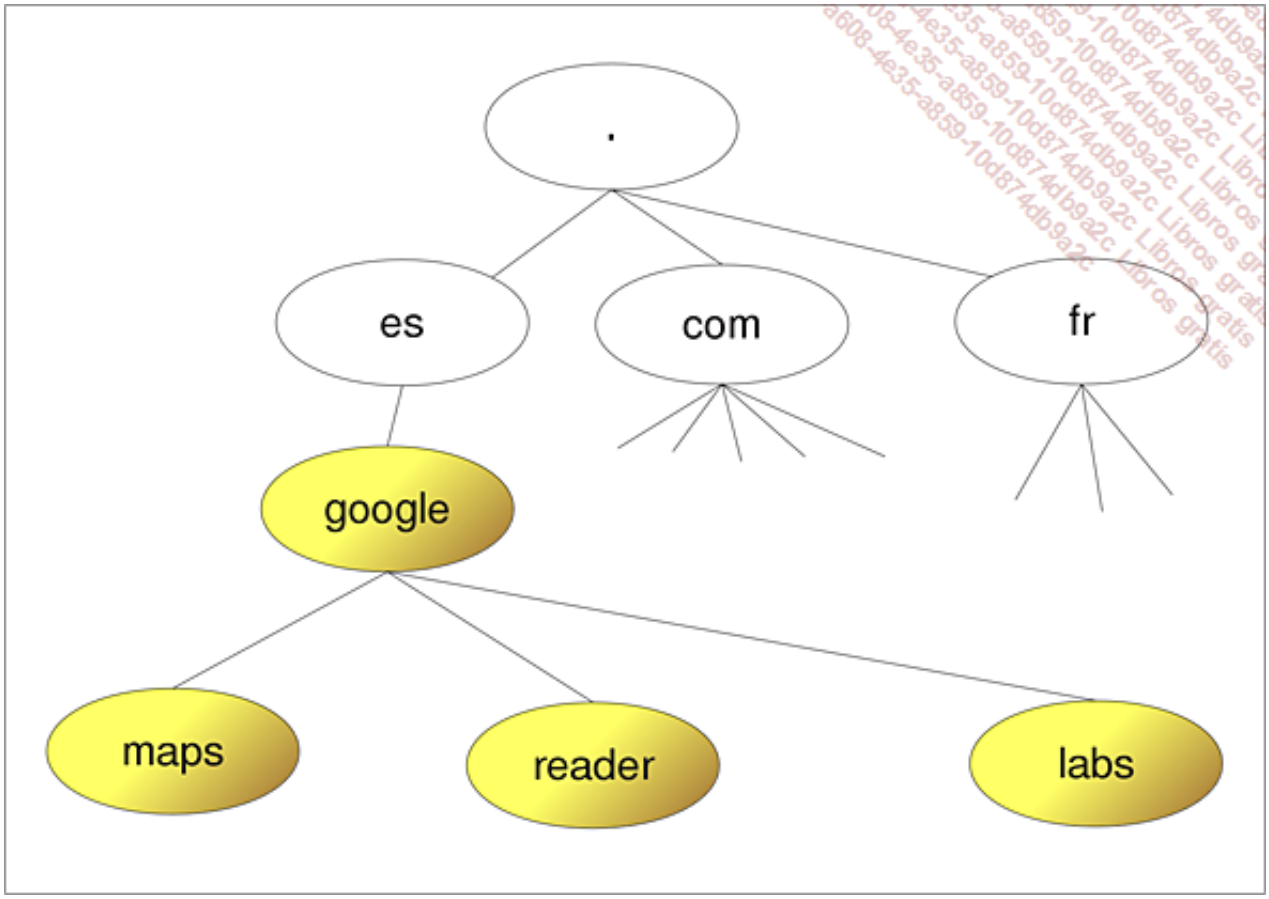
El DNS es seguro

Esta máxima apareció más tarde y no está todavía implementada en todos los servidores DNS. Sin embargo, ahora se pueden asegurar de extremo a extremo las operaciones DNS. Los servicios de seguridad disponibles son la autentificación, el control de acceso y el control de integridad.

2. Concepto de zonas DNS

El desorbitado número de registros DNS no permitiría su gestión sin un método de organización (sería como volver a tener un archivo **hosts** que albergara millones de líneas). Por lo tanto, se volvió indispensable su organización jerárquica y ésta es la razón de la existencia de las zonas DNS. Cada nivel de la jerarquía es una zona. Cada árbol es un dominio.

Se ha creado de forma arbitraria una zona llamada "." (punto), que es la raíz de la jerarquía y que contiene todos los **tld** (*top level domain*, dominio de nivel superior). Los **tld** son las archiconocidas extensiones tales como **com**, **es**, **net**, **fr**, etc. Todos los dominios que conocemos y utilizamos son subárboles de los **tld**.



En el ejemplo mostrado previamente, la zona google contiene las subzonas maps, reader y labs. Asimismo, también se puede decir que la zona "." contiene las subzonas es, com y fr. Las zonas situadas jerárquicamente por debajo de una zona se llaman zonas "hijo".

El interés de este tipo de organización es dedicar un servidor (de hecho por lo menos dos, por motivos de tolerancia a fallos) a la gestión de una zona. Como la jerarquía DNS es virtualmente ilimitada, tanto en anchura como en profundidad, un servidor DNS sólo gestiona una pequeña porción del espacio de nombres. Según nuestro ejemplo, si un servidor DNS alberga los datos de la zona google, a él irán dirigidas todas las consultas para la resolución de nombres acabados en "google.es". Sin embargo, no es necesario que albergue los datos de las zonas maps, reader y labs, debido a que le basta con redirigir la petición al servidor de la zona hijo. Se habla entonces de**delegación** en el sentido que se delega la gestión de una zona hijo a otro servidor.

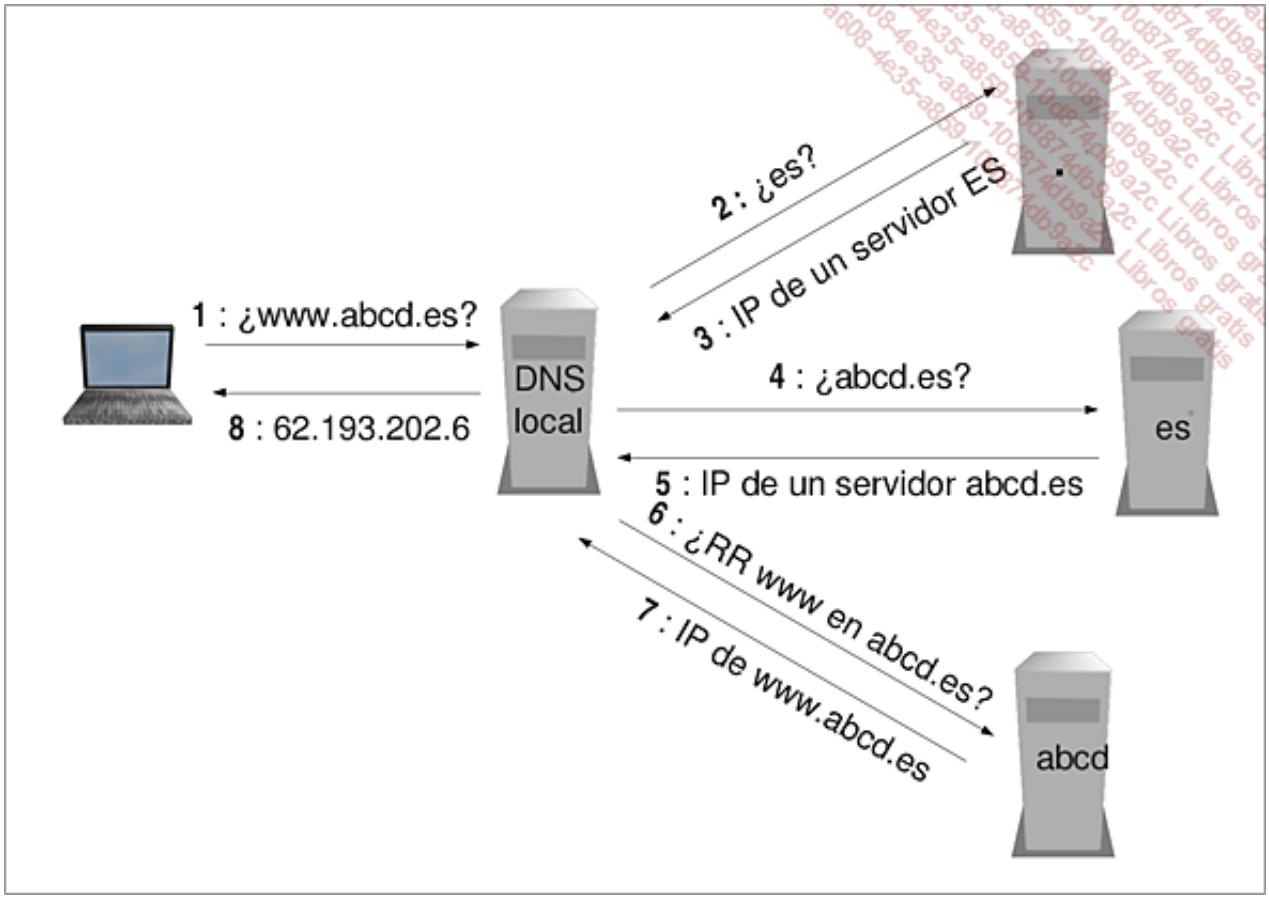
Por razones de tolerancia a fallos, los datos de cada zona DNS se deben replicar por lo menos una vez, es decir, tienen que existir al menos dos copias. Un servidor tendrá autoridad en la zona y será responsable de las actualizaciones. Se dice que es el **SOA** (*Start Of Authority*). Las zonas que alberga este servidor son de tipo **master** y los que albergan una réplica de la zona se configuran como **slave**.

3. Funcionamiento de la resolución de nombres

Cuando una aplicación tiene que hacer una resolución de nombres, se dirige al componente**resolver** de su sistema operativo. Entonces el **resolver** envía una petición de resolución de nombres al servidor DNS al que se hace referencia desde esta máquina. Las peticiones de cliente a servidor utilizan el puerto 53 y el protocolo de transporte UDP.

Si el servidor al que se le ha preguntado tiene localmente la información, responde directamente. Se dice que hace una respuesta **authoritative** (autoritaria).

Si le servidor al que se le ha preguntado no tiene la información, consultará la única zona que conoce, la zona ".", que le proporcionará la dirección de uno de los 13 servidores raíz de Internet. El servidor preguntará entonces a este servidor raíz para conocer la dirección de un servidor de la zona del **tld**. A este nuevo servidor se le preguntará por la dirección de un servidor de nombres que gestione la zona directamente por debajo del tld. Finalmente, a este siguiente servidor se le interrogará para saber si dispone del registro deseado en este dominio.



Esquema simplificado de la resolución de nombres:

1. El cliente pregunta al servidor al que hace referencia su configuración (proveedor de acceso o servidor local): ¿cuál es la dirección para el nombre **www.abcd.es**?
2. El servidor local solicita a un servidor raíz: dame la dirección de un servidor que conozca la zona **es**.
3. Éste, el servidor con dirección **193.176.144.6**, podrá informarte. Tiene toda la información de la zona **es**.
4. El servidor local solicita al servidor de la zona **es**: dame la dirección de un servidor que conozca la zona **abcd.es**.
5. Este servidor, cuya dirección es **213.41.120.195**, podrá informarte.
6. El servidor local pregunta al servidor de la zona **abcd.es**: ¿tienes un registro **www**en tu dominio **abcd.es**?
7. Sí, su dirección IP: **62.193.202.6**.
8. El servidor local a la estación cliente: me has solicitado **www.abcd.es** y su dirección IP es **62.193.202.6**.

4. Registros

Las zonas sólo tienen una función estructural, para proporcionar la resolución de nombres habrá que crear registros que harán corresponder un nombre a una dirección IP o a otro dato. Estos registros se llaman **Resource Records** (registros de recursos), a menudo escritos como **RR**, y constituyen la información fundamental de un DNS.

El **FQDN** (*Fully Qualified Domain Name*, nombre de dominio completamente cualificado) representa el nombre del host, con todo su árbol jerárquico, hasta la zona ".". Por ejemplo,**www.sanmarcelino.es** representa el registro **www** en la zona **sanmarcelino.es**, siendo **es** la última zona antes de la zona punto. Cuando se quiere escribir un nombre sin ningún tipo de ambigüedad en su interpretación como nombre DNS, se utiliza el **FQDN** con la zona punto presente, es decir, que se escribe un punto como último carácter del **FQDN**. Por lo tanto, se obtiene **www.sanmarcelino.es.**Esta notación es común e indispensable en los archivos de configuración DNS.

El objetivo primordial del sistema DNS es proporcionar un servicio de resolución de nombres. Es decir, crear la correspondencia entre un nombre de host y una dirección IP. Sin embargo, sus creadores previeron que el

sistema DNS sería capaz de proporcionar la resolución para otros tipos de nombres mejorando, de este modo, la finura del servicio.

a. Registros de tipo A

El más fácil de aprender y el más común. Es el registro que hace corresponder una dirección IP a un nombre. Por ejemplo, cuando se teclea http://www.sitio.es, www es un registro de tipo A en la zona sitio.es. Corresponde a una dirección IP que es la del servidor web que alberga el sitio en cuestión.

Resoluciones en la zona dominio.es

www → 82.25.120.5

soporte → 125.12.43.2

vpn → 82.25.120.6

b. Registros de tipo AAAA

Reciente pero cada vez más usado. Este registro hace corresponder una dirección IPv6 a un nombre.

Resoluciones en la zona dominio.es

www → 2001:610:12:123a:28:15ff:fed9:97e6

suporte → 2001:610:12:123a:28:15ff:fed9:97e8

c. Registros de tipo PTR

Pointer, justamente lo contrario de A. Si los registros de tipo A hacer corresponder una dirección IP a un nombre de host, los PTR hacen justo lo contrario. Existen en zonas un poco particulares llamadas IN-ADDR.ARPA.

El nombre estándar de la zona estará formado por los bytes de la parte de red de la dirección IP ordenados en sentido inverso, seguidos de la cadena de caracteres ".in-addr.arpa".

Resoluciones en la zona 1.168.192.in-addr.arpa

10 → servidor1.empresa.local (para servidor1.empresa.local → 192.168.1.10)

15 → printer1.empresa.local (para printer1.empresa.local → 192.168.1.15)

Resoluciones en la zona 85.in-addr.arpa

25.8.92 → www.abcd.es (para www.abcd.es → 85.92.8.25)

29.123.65 → www.def.net (para www.def.net → 85.65.123.29)

d. Registros de tipo CNAME

Canonical Name (alias o sobrenombre). Este tipo de registros hace corresponder un nombre a un nombre. Por ejemplo, si crea un servidor web para uso interno de su empresa en un servidor existente que se llamaría "produccion1.mibuzon.com", puede crear el CNAME "intranet" que es más intuitivo para los usuarios.

Resoluciones en la zona mibuzon.com

intranet → produccion1

impresora1 → printer1

e. Registros de tipo MX

Mail Exchanger (indicador de servidor de mensajería para un dominio). Este tipo de registros hace que los agentes de transferencia de correo sepan cuál es el servidor destinatario final de un correo. El ejemplo mostrado a continuación tiene fines ilustrativos y no presenta el formato de un registro MX.

Resolución en la zona dominio.es

@dominio.es → smtp.dominio.es → 82.25.120.6

f. Registro de tipo SOA

Start Of Authority (inicio de autoridad). Indica que el servidor tiene la responsabilidad de la zona. Toda zona en funcionamiento tiene un registro SOA.

Resolución en la zona dominio.es

dominio.es → ns.albergue.net

g. Registro de tipo NS

Name Server (servidor de nombres). Indica los servidores de nombres para la zona. Toda zona en funcionamiento tiene por lo menos un registro NS.

Resolución en la zona dominio.es

dominio.es → ns.albergue.net

5. DNS en Linux

a. El servidor DNS

Los servicios DNS que se ejecutan en Linux se basan, casi con total exclusividad, en el software**BIND** (*Berkeley Internet Name Domain*). Como su nombre indica, se creó en la universidad de Berkeley en California. Los primeros desarrollos tienen fecha de los años 80 y su mantenimiento lo realiza actualmente el ISC (*Internet System Consortium*), una asociación no lucrativa que gestiona un cierto número de programas estructurales de Internet y de redes locales.

Aunque existen alternativas al uso de BIND para la resolución de nombres en Linux (maradns y djbdns por ejemplo), solamente se exige el conocimiento de BIND en la certificación LPI.

b. El cliente DNS

Las máquinas Linux disponen de forma nativa de un cliente DNS llamado **resolver**. Toda aplicación que esté funcionando en Linux y necesite realizar una petición DNS utilizará este componente.

Este componente usa el archivo de configuración **/etc/resolv.conf**, que tiene una estructura muy sencilla.

Formato simplificado del archivo /etc/resolv.conf

```
search dominio
domain dominio
nameserver A.B.C.D
```

Archivo /etc/resolv.conf: directivas y variables utilizadas	
search	Opcional: indica el sufijo de búsqueda que hay que emplear en la máquina Linux. Evita tener que escribir la totalidad del FQDN en las aplicaciones. El archivo /etc/resolv.conf admite varios dominios de búsqueda especificados con search.
domain	Opcional y obsoleto: expresa el sufijo de búsqueda empleado en la máquina Linux.
dominio	El FQDN del dominio que forma el sufijo de búsqueda.
nameserver	Indica la dirección IP del servidor DNS que proporcionará las resoluciones. El archivo /etc/resolv.conf admite varios servidores DNS especificados por nameserver.

Configuración básica del servidor

1. Funcionamiento del servidor BIND

El servidor DNS BIND se basa en un ejecutable **named** y en un archivo de configuración **named.conf**.

a. Estructura del archivo named.conf y sus principales elementos de configuración

A continuación se muestra un ejemplo genérico del archivo **named.conf**. Según el caso, se puede encontrar en una forma completa y monolítica, pero es frecuente encontrarlo disperso en varias partes por razones de legibilidad. Los subarchivos se llaman mediante la directiva **include**. La función principal de este archivo es declarar las zonas que administrará este servidor y especificar todos los elementos de configuración.

Formato simplificado de named.conf

```
include "/ruta/archivo";
options {
        directory "/ruta/directoriodetrabajo";
        forwarders { A.B.C.D };
        };
zone "NOMBREDEZONA1" {
        type tipo;
        file "/RUTA/NOMBREARCHIVO1";
        };
zone "NOMBREDEZONA2" {
        type tipo;
        file "/RUTA/NOMBREARCHIVO2";
        };
```

Archivo named.conf: directivas principales utilizadas	
include	Indica el nombre de un "subarchivo" de configuración. Evita tener un archivo named.conf demasiado grande para administrarlo cómodamente.
options	Contenedor para algunas palabras clave, especialmente directory y forwarders.
directory	Se usa en directivas option. Indica el directorio utilizado para almacenar en disco datos de caché del servidor.
forwarders	Ubicado en una directiva option para las configuraciones simples (redirección incondicional). Si el servidor no dispone en sus archivos de la resolución solicitada, devuelve la petición al servidor cuya dirección IP es la dada junto a esta directiva.
zone	Contenedor para el nombre de una zona DNS administrada por el servidor.
type	Se usa en directivas zone. Indica el tipo de zona almacenada. Los valores principales son hint (servidores raíz), master (servidor maestro de una zona) y slave (réplica de un servidor master).
file	Se usa en directivas zone. Indica el archivo que contendrá la información de zona.

b. Archivos de definición de zona preinstalados

Según la implementación, algunas zonas se proporcionan por defecto en la instalación del servidor para habilitar un funcionamiento estándar y permitir las resoluciones de nombre comunes. Por ejemplo, la zona localhost que permite resolver el nombre localhost a 127.0.0.1 incluida en el interior del servicio DNS y no sólo en el archivo hosts.

Estos archivos de zona se crean durante la instalación y se les hace referencia en el archivo**named.conf**.

Ejemplo de archivo named.conf en una distribución Debian:

Observe que se incluye la declaración de las zonas por defecto así como la llamada a dos subarchivos de configuración llamados mediante la directiva include.

```
include "/etc/bind/named.conf.options";

zone "." {
  type hint;
  file "/etc/bind/db.root";
};

zone "localhost" {
  type master;
  file "/etc/bind/db.local";
};

zone "127.in-addr.arpa" {
  type master;
  file "/etc/bind/db.127";
};

zone "0.in-addr.arpa" {
  type master;
  file "/etc/bind/db.0";
};

zone "255.in-addr.arpa" {
  type master;
  file "/etc/bind/db.255";
};

include "/etc/bind/named.conf.local";
```

Observe que las directivas **include** devuelven dos archivos vacíos en la instalación (sólo contienen comentarios). El resto de la configuración finaliza con la declaración de las zonas, de las cuales la única realmente requerida para la resolución de nombres públicos es la zona ".", que está escrita en primer lugar.

2. Servidor de caché

Un servidor DNS de caché proporciona también la resolución de nombres, pero no contiene ningún dato de resolución local y requiere una infraestructura ya existente. Se limita a reenviar las peticiones a otros servidores. De este modo, el servidor pondrá en caché durante un tiempo determinado todas las resoluciones que ha reenviado.

Por definición, un servidor de caché no dispone localmente de zonas DNS personalizadas. Es decir, que no proporcionará por sí mismo ninguna resolución de tipo "¿Qué dirección IP le corresponde al nombre www.sitiogenial.com?": No guarda este tipo de información y deberá responder a las peticiones dirigiéndose a otros servidores mejor informados.

a. Configuración del servidor de caché

Ésta es la buena noticia: un servidor BIND recién instalado es de forma natural un servidor de caché. Por lo tanto no hay que realizar ningún tipo de configuración particular. Cuando se habla de instalar y configurar un servidor de caché como uno de los objetivos de la certificación LPI, se trata simplemente de instalar un servidor en funcionamiento sin información de zona local.

b. Redirección

Como ya se ha dicho, un servidor de caché no alberga de forma local registros de recursos. Si tiene que hacer una resolución, se dirigirá a los únicos servidores que conoce, que son los servidores raíz. Este método de resolución obviamente no es el más rápido y se podría desear aprovechar la caché de servidores ya en

funcionamiento, como los de un servicio de hospedaje o de un ISP. Para ello, hay que indicar a nuestro servidor la dirección de otros servidores a los que podrá redirigir sus peticiones. Este tipo de redirección se llama incondicional, ya que se redirigen todas las resoluciones no pesadas.

Configuración de la redirección en named.conf

```
options {
    forwarders {
      A.B.C.D;
    };
};
```

Archivo named.conf: directivas utilizadas para la redirección	
options	Anuncia la sección option en el archivo named.conf. Las redirecciones condicionales se establecen en una sección options.
forwarders	Se usa en directivas options. Establece la dirección o las direcciones IP del redirector o redirectores.

3. El comando de control rndc

Como todos los servicios Unix o Linux, BIND se inicia o se para mediante un script que está en **/etc/init.d**. Para gestionar al detalle el servicio, hay disponible un comando de control: **rndc**. Este comando junto con algunas palabras clave permite transmitir al servidor una gran variedad de instrucciones.

No es obligatorio usar **rndc** en la administración del día a día. Pero entonces cualquier modificación realizada en el archivo de configuración requerirá el reinicio completo del servicio y, por tanto, su interrumpción temporal. De este modo, **rndc** debería usarse sistemáticamente, sobre todo si el servidor gestiona una gran cantidad de zonas, como el caso de un alojamiento web, por ejemplo.

Sintaxis

```
rndc accion [parámetro]
```

Comando rndc: posibles acciones	
reload	Recarga los archivos de configuración y la información de zona.
reload zone*zona*	Recarga los archivos de una sola zona.
reconfig	Carga los archivos de configuración solamente para las nuevas zonas.
flush	Borra la caché del servidor.
flush *zona*	Borra la caché del servidor para una zona específica.
status	Muestra el estado del servidor.

Administración de zonas DNS

1. Administración de zonas locales

a. Creación de un archivo de zona directa

La información necesaria para la resolución deberá encontrarse en un archivo de declaración de zona. La ubicación de este archivo es libre, ya que se define en una sección **zone** de **named.conf**. Sin embargo, se ha establecido por un uso cotidiano que este archivo se ubique en el directorio **/var/named**. Tenga en cuenta que, según la distribución, también puede encontrarse en el directorio **/etc** o en **/etc/bind**. Para la certificación LPI, mejor recuerde **/var/named**.

Este archivo tiene un formato muy estricto que se muestra a continuación. En la mayoría de los casos, un fracaso en el inicio se debe a un archivo de zona mal formado. Se compone de las declaraciones de tiempo de vida en caché de los datos, de los servidores de nombres de esta zona y del conjunto de registros de recursos (RR) de esta zona.

Formato típico del archivo de zona directa

```
$TTL      ttl
nombrezona  IN  SOA  servidor mailadmin (
         serial
         refresh
         retry
         expire
         negative )

nombrezona  IN  NS  servidor
```

Archivo de zona directa: formato típico de la cabecera	
ttl	*Time To Live* (tiempo de vida): indica la duración en segundos que se conservarán los datos en memoria caché. Este valor se precede por la directiva $TTL.
nombrezona	FQDN de la zona administrada en este archivo. A menudo reemplazado por una arroba (@) para aligerar el archivo. Atención, como se trata de un FQDN, el nombre de la zona debe acabar con un punto.
IN	Obsoleto a la vez que actual: clase Internet (no hay otra clase que se pueda usar).
SOA	*Start Of Authority*. Registro obligatorio para indicar que este servidor es legítimo en esta zona.
servidor	FQDN del servidor que tiene autoridad en la zona.
mailadmin	Dirección de correo del administrador del servidor. La arroba es un carácter reservado en los archivos de zona, convencionalmente se reemplaza por un punto. admin@midominio.es pasaría a ser entonces admin.midominio.es.
serial	Valor numérico. Número de serie del archivo. Útil cuando la zona se replica en otros servidores para saber si los datos han cambiado y si debe hacerse la réplica.
refresh	Valor numérico. Utilizado cuando la zona se replica. Indica al servidor esclavo con qué intervalo comprobar la validez de su zona.
retry	Valor numérico. Utilizado cuando la zona se replica. Si es imposible para el servidor esclavo contactar con el servidor maestro, indica cuánto esperar antes de volverlo a intentar.
expire	Valor numérico. Utilizado cuando la zona se replica. Si es imposible para el servidor esclavo contactar con el servidor maestro, indica con

	cuánto tiempo los registros sin refrescar pierden su validez y deben dejarse de usar.
negative	Valor numérico. Indica cuánto tiempo el servidor debe conservar en caché una respuesta negativa.
NS	Registro que indica cuál es el servidor de nombres para esta zona.

b. Creación de un archivo de zona inversa

El archivo de zona inversa tendrá la misma estructura que un archivo de zona directa. Como se ha indicado anteriormente, el nombre normalizado de la zona se forma con los bytes de la IP de partida ordenados en sentido inverso seguidos de la cadena de caracteres ".in-addr.arpa". Por ejemplo, la zona inversa para la red **192.168.99.0** será: **99.168.192.in-addr.arpa** y éste es el nombre que deberá usarse en el archivo de zona y en el archivo **named.conf**.

Formato típico de un archivo de zona inversa

```
$TTL    ttl
nombrezonainv  IN  SOA  servidor mailadmin (
         serial
         refresh
         retry
         expire
         negative )

nombrezonainv  IN  NS  servidor
```

Archivo de zona inversa: formato típico de la cabecera	
nombrezonainv	Nombre normalizado de la zona inversa: *subred_invertida*.in-addr.arpa. Donde *subred_invertida* representa los bytes de la subred en orden inverso. Atención: el nombre de la zona inversa es un FQDN, por lo tanto tiene que terminar con un punto.
SOA	*Start Of Authority*. Registro obligatorio para indicar que este servidor es legítimo en esta zona.
servidor	FQDN del servidor que tiene autoridad en la zona.
NS	Registro que indica cuál es el servidor de nombres para esta zona.

Como se puede comprobar es exactamente lo mismo que para la zona directa. Es el formato de los registros los que marcan las diferencias.

c. Creación de registros en los archivos de zona

Una vez que los archivos de zona se han creado, basta con añadir tantos registros de recursos como se desee, a razón de uno por línea.

Formato de un registro de recurso en un archivo de zona directa

```
nombre IN tipoRR valor_resuelto
```

Formato de un registro de recurso en un archivo de zona inversa

```
dirección_host IN PTR nombre
```

Archivo de zona: formato de los registros	
nombre	Nombre simple o FQDN al que se le crea la correspondencia con una dirección IP.
IN	Obsoleto pero necesario: clase Internet.
tipoRR	Tipo de registro. A menudo de tipo A: crea la correspondencia entre una IP y un nombre. Valores comunes: A, CNAME, MX.

valor_resuelto	Es lo que se hace corresponder a un nombre. En el caso de un registro de tipo A, se trata de una dirección IP.
dirección_host	El byte o bytes que están asociados a la dirección de red de la zona inversa formarán la dirección IP que se resolverá.
PTR	Tipo puntero: crea la correspondencia de un nombre con una dirección IP. Aparte de los registros SOA y NS, son los únicos que se encuentran en las zonas inversas.

Evidentemente, añadir un gran número de registros es una tarea tediosa. Es mucho mejor hacerlo mediante un script.

Ejemplo de script simple para rellenar un archivo de zona:

Los alojamientos y otros DNS que gestionan un gran número de registros utilizan de forma natural scripts más elaborados.

```
#!/bin/bash
echo "¿Nombre que se añadirá a la zona?"
read nombre
echo "¿Dirección IP correspondiente?"
read ip
echo "$nombre  IN  A  $ip" >> /var/named/midominio.es
```

d. Declaración de una zona principal en el archivo named.conf

Una vez que se dispone de un archivo de zona, hay que darlo a conocer al servidor que debe cargarlo en el arranque. Esto se realiza mediante una declaración normalizada en el archivo**named.conf**.

Formato típico de la declaración de una zona en named.conf

```
zone "nombrezona" {
    type master;
    file "archivo";
};
```

Archivo named.conf: directivas y sintaxis de la declaración de zonas	
nombrezona	El FQDN de la zona gestionada por el servidor.
type master	Determina que se trata de una zona maestra que se debe sincronizar con posibles servidores esclavos.
archivo	Ruta absoluta del archivo que se debe leer para conocer todos los elementos propios de la zona (configuración, RR, etc.).

e. Actualizar la nueva configuración

A continuación hay que hacer que el servidor recargue los archivos de configuración para que tenga en cuenta los cambios realizados. Para ello, tenemos dos alternativas: el reinicio del servicio o la carga de la nueva configuración mediante el comando de control **rndc**.

Reinicio del servicio

```
/etc/init.d/bind9 restart
```

Carga de la nueva zona mediante rndc

```
rndc reload midominio.es
```

2. Gestión de zonas secundarias

Una zona DNS no debería depender de un único servidor y por ello es frecuente crear un segundo servidor de zonas secundarias, estrictamente idénticas a las zonas primarias y sincronizadas a intervalos regulares.

a. Declaración de la zona secundaria en named.conf

Evidentemente no es necesario crear los archivos de zona, ya que se sincronizarán desde el servidor autoritario. Se habla comúnmente de servidor maestro y servidores esclavos.

La carga de la zona esclava se realiza mediante una declaración de zona normalizada en el archivo**named.conf**.

Formato típico de la declaración de zona secundaria en el archivo named.conf

```
zone "nombrezona" {
    type slave;
    masters { dirección_maestro; };
    file "archivo";
};
```

Archivo named.conf: directivas y sintaxis de la declaración de zona	
nombrezona	El FQDN de la zona gestionada por el servidor.
type slave	Determina que se trata de una zona esclava que se tendrá que sincronizar desde un servidor maestro.
dirección_maestro	Dirección IP del servidor autoritario.
archivo	Ruta absoluta al archivo en el que se almacenan los elementos sincronizados. La cuenta de servicio tiene que tener los permisos de escritura habilitados en el directorio de trabajo.

b. Consideración de la nueva configuración

A continuación hay que hacer que el servidor recargue los archivos de configuración para que tenga en cuenta los cambios realizados. Para ello, tenemos dos alternativas: el reinicio del servicio o la carga de la nueva configuración mediante el comando de control **rndc**.

Reinicio del servicio

```
/etc/init.d/bind9 restart
```

Carga de la nueva zona mediante rndc

```
rndc reload midominio.es
```

3. Delegación de zona

Una delegación de zona consiste en hacer que un servidor de terceros gestione una zona hija albergada por un servidor padre. Es el principio de la delegación el que permite distribuir el conjunto de espacio de nombres DNS en miles de servidores. La delegación se configura en el servidor padre.

Para ello se añaden dos **Resource Records** en el archivo de zona del padre: uno de tipo **NS** para indicar que existe un servidor de nombres para la zona hija y otro de tipo **A** para saber la dirección IP de este servidor de nombres. El registro **NS** que proporciona la delegación se llama **glue record**(registro de pegado).

Configuración de la delegación en el archivo de la zona padre

```
zona_hija IN NS dns_hijo
dns_hijo IN A A.B.C.D
```

Elementos	
zona_hija	Nombre simple de la zona hija.
IN	Obsoleto pero obligatorio: clase Internet.
NS	Este registro es de tipo Name Server (servidor de nombres).
dns_hijo	Nombre del servidor DNS que administra la zona hija.

A	Es un registro de tipo A.
A.B.C.D	Dirección IP del servidor de nombres para la zona hija.

4. Herramientas de comprobación

a. ping

Aunque no es su función principal, **ping** se puede usar como prueba rudimentaria para la resolución de nombres. Entonces se limitará a comprobar la respuesta de los servidores por defecto, que se informan en **/etc/resolv.conf**.

Utilización de ping para comprobar resoluciones de nombres

Cuando se utiliza ping para comprobar la resolución de nombres, es la traducción de la dirección lo que importa, no la respuesta ICMP de la máquina remota.

```
donald:/etc/bind# ping donald.formacion.es
PING donald.formacion.es (192.168.1.1) 56(84) bytes
64 bytes from donald.formacion.es (192.168.1.1): icmp
64 bytes from donald.formacion.es (192.168.1.1): icmp
64 bytes from donald.formacion.es (192.168.1.1): icmp
```

b. nslookup

nslookup es la herramienta más popular de consulta a servidores DNS. Está disponible en la gran mayoría de plataformas Unix y Windows.

nslookup se utiliza en la mayoría de las veces en modo interactivo. Es decir, después de haber teclado **nslookup** se accede a su interfaz donde se introducen comandos específicos. Los servidores de nombres consultados por defecto son aquellos que se hace referencia en**/etc/resolv.conf**. Esto puede cambiar si así se desea a través de esta herramienta.

Utilización de nslookup para una resolución de nombres

Por defecto, nslookup dirige a los servidores DNS consultas de tipo A.

```
donald:/etc/bind# nslookup
> server
Default server: 192.168.1.1
Address: 192.168.1.1#53
> cuacua.formacion.es
Server:         192.168.1.1
Address:        192.168.1.1#53

cuacua.formacion.es   canonical name = donald.formacion.es.
Name:   donald.formacion.es
Address: 192.168.1.1
>
```

Peticiones y parámetros en la interfaz interactiva de nslookup	
nombre	Teclear un nombre DNS directamente en la interfaz de nslookup equivale a solicitar su resolución. nslookup indica entonces a qué servidor DNS ha consultado y la respuesta que le ha dado. Se puede tratar de un nombre completo (FQDN) o de uno simple si se basa en un sufijo de búsqueda definido en /etc/resolv.conf.
server *A.B.C.D*	El comando server seguido de la dirección IP de un servidor indica a nslookup que todas las futuras consultas se deberán dirigir a este servidor.
set type=*TIPO*	Por defecto, nslookup realiza consultas de tipo A (resolución normal

Utilización de nslookup para encontrar la dirección de un servidor de correo

nslookup se puede

	de nombres en direcciones IPv4). El comando set type permite realizar consultas de otro tipo. Se usa comúnmente para saber, por ejemplo, los servidores de nombres o de correo asociados a una zona.

usar para todo tipo de registros comunes

(en este caso MX).

```
donald:/etc/bind# nslookup
> set type=MX
> marte.org
Server:         192.168.1.1
Address:        192.168.1.1#53

Non-authoritative answer:
marte.org      mail exchanger = mail.marte.org

Authoritative answers can be found from:
>
```

c. dig

dig es la nueva herramienta propuesta por el ISC para la consulta y el diagnóstico de servidores DNS. Llegando a ser la más precisa y exitosa de las herramientas de test, con el tiempo acabará siendo la solución de referencia. Sin embargo, el hábito de los administradores de DNS todavía sugiere un brillante futuro para nslookup.

dig se utiliza en modo no interactivo, es decir, cada vez que se usa **dig** se le debe proporcionar los parámetros necesarios para la resolución.

Sintaxis simplificada de dig

```
dig nombre
dig A.B.C.D nombre TIPO
```

Elementos	
nombre	El nombre completo (FQDN) del que se quiere obtener la resolución.
A.B.C.D	La dirección IP del servidor DNS al que se consultará. En caso de omisión, se consultarán los servidores a los que se hace referencia en /etc/resolv.conf.
TIPO	Por defecto, dig realiza consultas de tipo A (resolución normal de nombres con direcciones IPv4). Si se usa el parámetro TIPO, permite definir otros tipos de búsqueda con el objetivo de saber, por ejemplo, los servidores de nombres o de correo asociados a una zona.

Ejemplo de utilización de dig

Con diferencia se trata del comando más preciso de

diagnóstico DNS.

```
donald:/etc/bind# dig @127.0.0.1 cuacua.formacion.es

; <<>> DiG 9.2.4 <<>> @127.0.0.1 cuacua.formacion.es
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 18067
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 1, ADDITIONAL: 0

;; QUESTION SECTION:
;cuacua.formacion.es.         IN      A

;; ANSWER SECTION:
cuacua.formacion.es.    86400   IN      CNAME   donald.formacion.es.
donald.formacion.es.    86400   IN      A       192.168.1.1
```

```
;; AUTHORITY SECTION:
formacion.es.           86400   IN      NS      donald.formacion.es.

;; Query time: 9 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Thu Jun 15 19:49:45 2006
;; MSG SIZE  rcvd: 90
```

d. host

host es una sencilla herramienta para realizar peticiones DNS en modo no interactivo.

Sintaxis simplificada del comando host

```
host nombre

host nombre -t tipo A.B.C.D
```

Elementos	
nombre	El nombre DNS del que se desea realizar la resolución. Se puede tratar de un FQDN o de un nombre simple que se completará con el sufijo de búsqueda si se ha definido en /etc/resolv.conf.
-t *tipo*	Opcional: el tipo de consulta que se solicita. Por defecto el tipo se selecciona automáticamente de entre los tipos A, AAAA y MX.
A.B.C.D	Opcional: la dirección IP del servidor DNS al que se le realiza la consulta. Si este elemento no se informa, serán los servidores en /etc/resolv.conf los que se utilizarán.

Utilización de host para

comprobar una resolución de nombres

host devuelve un resultado escueto.

```
donald:/etc/bind# host cuacua.formacion.es
cuacua.formacion.es. is an alias for donald.formacion.es.
donald.formacion.es has address 192.168.1.1

donald:/etc/bind#
```

Utilización de host para obtener los registros NS

```
donald:/etc/bind# host -t NS formacion.es
formacion.es name server donald.formacion.es.

donald:/etc/bind#
```

e. Medición de rendimiento

El comando **time**, que mide el tiempo consumido por una aplicación, permite medir el rendimiento de una resolución DNS. Indica el tiempo total consumido por el comando y el tiempo consumido por los procesos en el espacio de ejecución del sistema y en el del usuario.

Consulta del tiempo usado para una resolución DNS

Los tiempos medidos dependen del ancho de banda disponible, de la disponibilidad del servidor y de la rapidez de la máquina cliente.

```
usuario@servidor:~$ time nslookup www.ediciones-eni.com
Server:    127.0.0.1
Address:  127.0.0.1#53
```

```
Non-authoritative answer:
Name:  www.ediciones-eni.com
Address: 81.80.245.20


real  0m0.256s
user  0m0.000s
sys   0m0.010s
usuario@servidor:~$
```

Seguridad en el servicio DNS

1. Limitaciones de los clientes

Se pueden limitar las consultas permitidas. La directiva **allow-query** en el archivo de configuración permite definir los hosts o las redes a los que el servidor tiene permitido responder.

Limitación de clientes autorizados en el archivo named.conf

```
allow-query { redes_autorizadas; };
```

Donde *redes_autorizadas* representa la o las direcciones de red o de equipo que pueden dirigirse al servidor.

2. Utilización de una cuenta de servicio

a. ¿Por qué una cuenta de servicio?

En los orígenes, era frecuente ejecutar un servidor **bind** con la cuenta de administración **root**. Es decir, la cuenta **root** era propietaria del proceso. Las consecuencias podían llegar a ser desastrosas: si el ejecutable **named** enviaba código sensible (peligroso) al procesador, lo hacía en nombre de **root**, es decir con privilegios absolutos en el sistema. Esta situación presenta una serie de riesgos. Pueden ser errores en el código ejecutable de **named** o bien vulnerabilidades del programa que permitirían a un atacante enviar código ejecutable al procesador.

La solución es, en general, ejecutar **named** con unas credenciales distintas a las de **root**, utilizando una cuenta de servicio: una cuenta de usuario que no permita la conexión directa al sistema, pero que será propietaria del proceso. De este modo, si se ejecuta código malicioso proveniente del proceso **named**, no tendrá más privilegios que los de la cuenta de servicio y, por tanto, no pondrá en peligro el sistema.

La mayoría de las implementaciones modernas de **bind** usan de forma nativa una cuenta de servicio.

Cuenta de servicio named en una distribución Red Hat

La cuenta se crea automáticamente con la instalación del servicio.

```
[root@RH9 root]# grep named /etc/passwd
named:x:25:25:Named:/var/named:/sbin/nologin
```

b. Ejecución de named con una cuenta de servicio

Como ya se ha dicho, todas las implementaciones de **bind** utilizadas en las distribuciones modernas de Linux usan de forma nativa una cuenta de servicio. A continuación se muestra cómo se incluye esta forma de ejecución en los scripts de inicio del servicio.

Sintaxis simplificada del comando named para su uso con una cuenta de servicio

```
named -u usuario
```

Elementos	
named	El ejecutable principal de bind. En la mayoría de las implementaciones, ejecutado desde el script de gestión del servicio.
usuario	Llamado con el parámetro -u, indica la cuenta de servicio propietaria del proceso. Por supuesto, esta cuenta tiene que haberse definido en el archivo /etc/passwd.

3. Bind en modo chroot

a. ¿Para qué enjaular el proceso?

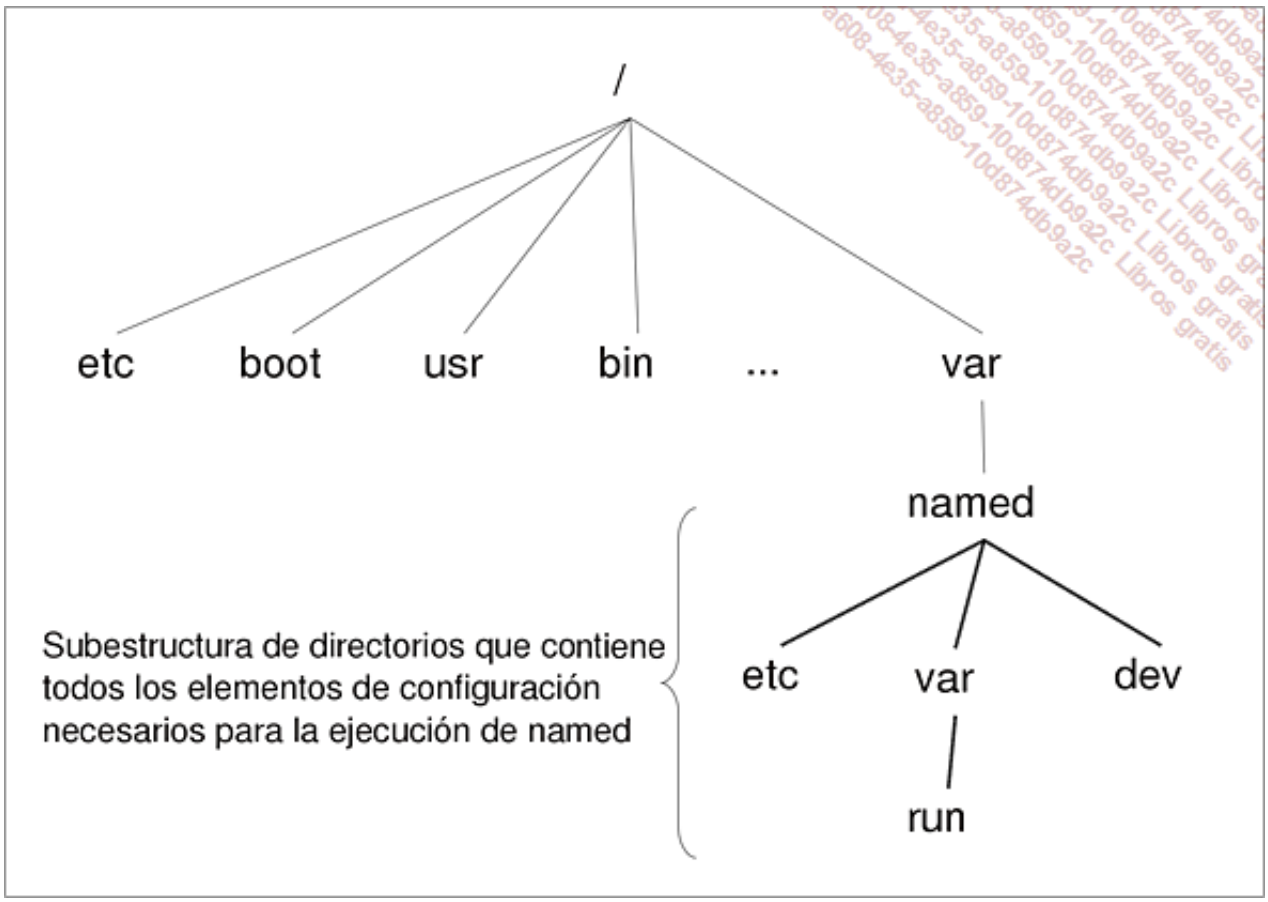
Se ha visto anteriormente que un uso malintencionado del proceso **named** podía entrañar el riesgo de ejecución de acciones peligrosas para el sistema. El enjaulamiento del proceso en un directorio dedicado permite limitar los riesgos. El objetivo es hacer creer al proceso que se está ejecutando en un sistema normal, mientras que realmente está enclaustrado en una estructura de directorios paralela y en ningún caso puede interactuar con el resto del sistema. El término "enjaulamiento" no ha sido usurpado, en inglés también se habla de poner "in jail", es decir, en prisión.

Entonces se dice que se está utilizando **bind** en modo **chroot**, contracción de **change root** (cambio de raíz).

Se recomienda utilizar el modo **chroot** con una cuenta de servicio. Un proceso que tuviera los privilegios de **root** podría obtener el modo de salir del directorio donde ha sido enjaulado.

b. Creación del entorno necesario

En la media en que se engaña al proceso y éste cree que se ejecuta en un entorno normal, debe tener a su disposición todos los elementos necesarios para su funcionamiento. Hay que entender que el proceso no tendrá ningún modo de ir a buscar cualquier tipo de dato fuera de su directorio. La puesta en marcha de un **bind** en modo **chroot** requiere, por tanto, una fase preliminar de creación de su entorno de trabajo.



Fases de creación del entorno de trabajo

- Creación del directorio de chroot.
- Creación de la estructura de directorios falsa "/" en el directorio de chroot. Todos los directorios utilizados por el proceso named deben aparecer ahí.
- Copia de los archivos de configuración en el directorio de chroot.
- Ejecución del proceso en modo chroot.

c. Ejecución del programa en modo chroot

Realmente no es difícil ejecutar **bind** en modo **chroot**.

Sintaxis del comando named para su uso en modo chroot

```
named -c config -u usuario -t directorio
```

Elementos usados en la sintaxis:	
named	El ejecutable principal de bind. En la mayoría de las implementaciones, ejecutado desde el script de gestión del servicio.
config	Opcional. Indica el archivo de configuración que se usará en la carga. En principio /etc/named.conf o /etc/bind/named.conf.
usuario	La cuenta de servicio propietaria del proceso. Por supuesto, esta cuenta tiene que definirse en el archivo /etc/passwd.
directorio	El directorio en el que se enjaulará named. A menudo, /var/named.

Es recomendable comprobar en los registros que el proceso consigue iniciarse correctamente en su nuevo entorno. En general, se requieren algunos intentos.

4. Intercambio seguro entre servidores

Muchas funciones de Internet requieren el uso de DNS, desde algo tan sencillo como navegar hasta el envío de correos. Los intentos de phishing, cada vez más frecuentes, muestran el peligro que entraña la incertidumbre en la resolución de nombres. Si alguien se conecta al sitio web de su banco en línea usando la url exacta pero el nombre se resuelve con la dirección IP de un farsante, las consecuencias pueden llegar a ser dramáticas. En el caso de los DNS, la seguridad se basa sobre todo en la autenticación y la integridad de los datos. Es decir, se quiere tener la certeza de que se está dialogando con el servidor adecuado y los datos no se han modificado en el trayecto.

En este caso se usará el mecanismo **TSIG** (*Transaction SIGnature*, firma de transacciones). Este mecanismo se basa en el uso de una clave compartida entre los servidores que intercambian datos.

a. Generación de la clave compartida

Existe una herramienta de generación de claves: **dnssec-keygen**. Tiene muchos posibles usos, pero el que se muestra a continuación es su uso para TSIG.

Sintaxis de dnssec-keygen en su uso para TSIG

```
dnssec-keygen -a HMAC-MD5 -b tamaño_de_clave -n nametype nombreclave
```

dnssec-keygen: variables y parámetros	
-a HMAC-MD5	-a define el algoritmo de encriptación. HMAC-MD5 es el único valor soportado para TSIG.
-b*tamaño_de_clave*	-b define el tamaño de la clave usada. Para HMAC-MD5,*tamaño_de_clave* tiene que estar comprendido entre 1 y 512. 128 es un valor corriente generalmente satisfactorio.
-n *nametype*	-n define la propiedad de la clave. En su uso para TSIG, generalmente *nametype* tiene el valor HOST para indicar que la seguridad va de máquina a máquina.
nombreclave	El nombre de la clave. Puede ser cualquier cadena alfanumérica.

El comando finaliza con la generación de dos archivos: **Knombreclave.+xxx.yyyyy.key** y**Knombreclave.+xxx.yyyyy.private**.

Ejemplo de uso de dnssec-keygen

```
donald:~# dnssec-keygen -a HMAC-MD5 -b 128 -n HOST supersecret
Ksupersecret.+157+26824

donald:~# cat Ksupersecret.+157+26824.key
supersecret. IN KEY 512 3 157
  yItYGlAQtGcM7VqGjZdJAg==
```

```
donald:~# cat Ksupersecret.+157+26824.private
Private-key-format: v1.2
Algorithm: 157 (HMAC_MD5)
Key: yItYGlAQtGcM7VqGjZdJAg==
```

b. Declaración de la clave en named.conf

En los servidores afectados por esta medida de seguridad, se incluirá en el archivo **named.conf** la definición de la clave.

Sintaxis de declaración de claves en named.conf

```
key nombreclave {
   algorithm hmac-md5;
   secret "yItYGlAQtGcM7VqGjZdJAg==";
};
```

Elementos usados en esta sintaxis:	
key	Inicia la declaración de la clave.
nombreclave	El nombre de la clave utilizada en la generación.
algorithm	Tiene como parámetro el tipo de algoritmo usado.
hmac-md5	Obligatorio para TSIG.
secret	Tiene como parámetro la clave generada en el archivo Knombreclave.+xxx.yyyyy.key (la cadena de caracteres entre comillas dobles).

c. Ambos servidores tienen que usar la clave

La clave compartida se declara en ambos servidores. Ahora hay que hacer que sepan que tienen que utilizarla para garantizar la seguridad de ciertas comunicaciones. Por lo tanto, habrá que añadir un nuevo comando en **named.conf**.

Sintaxis de uso de la clave en named.conf

```
server ip_dest {
   keys { nombreclave; };
};
```

Elementos empleados en esta sintaxis:	
server	Anuncia un modo de comportamiento para un servidor determinado.
ip_dest	La dirección IP del servidor para el que se aplica esta directiva.
keys	Establece la clave utilizada para asegurar los intercambios.
nombreclave	El nombre de la clave utilizada en la generación.

d. Rechazar todo servicio que no esté firmado

El párrafo anterior ha dado a los servidores la capacidad de comunicarse de forma segura. A continuación hay que hacer que esta medida de seguridad sea obligatoria para todas las peticiones entre servidores, por ejemplo en el contexto de una delegación.

Sintaxis

```
zone "midominio.es" {
    type master;
    file "db.midominio.es";
    allow-recursion { key supersecret; };
};
```

Comprobación de los conocimientos adquiridos: preguntas/respuestas

Ponga a prueba sus conocimientos respondiendo a las preguntas siguientes. Estas preguntas no siempre esperan una respuesta cerrada. Las preguntas planteadas en la certificación, a pesar de que abordan los mismos temas, en su mayoría serán planteadas en forma de preguntas de opción múltiple o que requieran una respuesta corta, en palabras escritas en el teclado.

1. Preguntas

1. ¿Por qué existen varios tipos de registros DNS?
2. ¿Qué relación existe entre la mensajería y la resolución DNS?
3. ¿En qué aspecto un servidor de caché mejora el funcionamiento global de una red?
4. ¿Cómo se instala el componente resolver DNS en un sistema Linux estándar?
5. ¿Se puede prescindir de la directiva include en el archivo named.conf?
6. ¿Cómo un servidor DNS local dentro de una empresa puede aprovechar la caché de un servidor más solicitado como el de su proveedor de servicios de Internet por ejemplo?
7. ¿Qué se puede hacer para que un servidor DNS actualice su configuración teniendo en cuenta nuevos elementos, como por ejemplo nuevos registros, sin tener que recurrir a un reinicio total del servicio?
8. De entre todos los comandos de comprobación de resolución DNS, ¿cuál proporciona los resultados más precisos y detallados?
9. ¿Cómo se puede proteger de una vulnerabilidad de código ejecutable DNS cuando se teme que se produzca una intrusión o un daño mediante el uso de esta vulnerabilidad?
10. En el contexto de una delegación, ¿por qué es indispensable tener disponibilidad de comunicación entre el servidor que delega y el servidor delegado?

2. Respuestas

1. ¿Por qué existen varios tipos de registros DNS?

   Para almacenar información de distinto tipo. De este modo, un cliente puede solicitar a su servidor DNS información precisa. Por ejemplo: ¿Cuál es el servidor maestro para esta zona? ¿Cuáles son los servidores DNS disponibles para esta zona?

2. ¿Qué relación existe entre la mensajería y la resolución DNS?

   Los registros MX hacen referencia a los servicios de mensajería en un dominio DNS. Un servidor de mensajería que esté funcionando puede que no reciba ningún mensaje del exterior si a su registro MX no se le hace referencia correctamente.

3. ¿En qué aspecto un servidor de caché mejora el funcionamiento global de una red?

   Conservando en memoria las resoluciones ya realizadas, el servidor responde mucho más rápidamente en las siguientes peticiones. En la medida en que en una red normal la mayoría de las peticiones se realizan sobre los mismos registros comunes (google, hotmail, ebay, etc.), todos los clientes obtienen una respuesta inmediata sin que el servidor tenga que volver a realizar una resolución pública.

4. ¿Cómo se instala el componente resolver DNS en un sistema Linux estándar?

   El resolver forma parte de la pila IP en todas las distribuciones Linux y, por lo tanto, no hay que instalarlo.

5. ¿Se puede prescindir de la directiva include en el archivo named.conf?

   Por supuesto. Esta directiva permite llamar a archivos que contengan elementos de configuración anexa e integrarlos en la configuración del servidor. Sin embargo, si se decide ubicar todos los elementos de configuración en un solo archivo named.conf, aunque no genere problemas, se obtendrá un archivo de configuración un poco largo.

6. ¿Cómo un servidor DNS local dentro de una empresa puede aprovechar la caché de un servidor más solicitado como el de su proveedor de servicios de Internet por ejemplo?

   Declarando una redirección hacia él (directiva forwarders). El servidor resolverá entonces todos los

registros que pertenezcan a zonas locales y se dirigirá al servidor de su proveedor de servicios de Internet para cualquier otra resolución.

7 ¿Qué se puede hacer para que un servidor DNS actualice su configuración teniendo en cuenta nuevos elementos, como por ejemplo nuevos registros, sin tener que recurrir a un reinicio total del servicio?

Gracias al comando de control rndc, que permite precisamente tener en cuenta los cambios en la configuración o en los datos albergados en una sola zona evitanto, por tanto, la recarga completa del servicio que requeriría muchos más recursos.

8 De entre todos los comandos de comprobación de resolución DNS, ¿cuál proporciona los resultados más precisos y detallados?

El comando dig, considerado el buque insignia de las herramientas de diagnóstico DNS. Mientras que algunas distribuciones tienen a dig como la herramienta universal destinada a reemplazar con ciertas ventajas a las demás, la mayoría de los usuarios siguen prefiriendo hoy en día el comando nslookup.

9 ¿Cómo se puede proteger de una vulnerabilidad de código ejecutable DNS cuando se teme que se produzca una intrusión o un daño mediante el uso de esta vulnerabilidad?

Mediante dos técnicas que se usan a menudo conjuntamente. Para empezar, enjaulando el proceso named en un entorno de ejecución impermeable. Se dice que se mete el proceso en prisión (in jail) mediante un cambio de raíz (chroot). La prisión en cuestión es una estructura de directorios que contiene una copia de todos los elementos que necesitará el proceso para funcionar, por ejemplo librerías o archivos de configuración. Además, se ejecuta el proceso con una cuenta de servicio con privilegios limitados, por lo que el uso de una posible vulnerabilidad no daría al atacante más permisos que los de la cuenta en cuestión.

10 En el contexto de una delegación, ¿por qué es indispensable tener disponibilidad de comunicación entre el servidor que delega y el servidor delegado?

El principio de una delegación es externalizar la administración de una zona hija usando otro servidor. Todas las peticiones que se hagan al servidor padre para la zona hija se dirigirán dinámicamente al servidor de la zona hija. Si no hay comunicación, esta redirección es simplemente imposible.

Trabajos prácticos

1. Instalación de un servidor DNS

Para acelerar la navegación por Internet en el interior de la red, decide desplegar un servidor DNS interno en su empresa. De este modo, todos los clientes que hagan las resoluciones DNS más comunes aprovecharán la caché del servidor local.

a. Instalación de los servicios software

En el servidor alfa, instale el servidor bind con el comando siguiente:

```
apt-get install bind9
```

El servicio deberá estar ya instalado en el servidor beta.

b. Verificación

Comandos útiles

- pgrep
- ps
- rndc

Operaciones

1. Comprobar que el servicio está en ejecución observando los procesos activos.
2. Comprobar que el servicio está en ejecución consultando el script de inicio.
3. Comprobar que el servicio está en ejecución usando el comando de control rndc.

Resumen de los comandos y resultado por pantalla

Observación del proceso:

```
alfa:~# pgrep -l named
4491 named
alfa:~#
```

Consulta del script:

```
alfa:~# /etc/init.d/bind9 status
bind9 is running.
alfa:~#
```

Utilización del comando de control:

c.

```
alfa:~# rndc status
version: 9.6-ESV-R4
CPUs found: 1
worker threads: 1
number of zones: 17
debug level: 0
xfers running: 0
xfers deferred: 0
soa queries in progress: 0
query logging is OFF
```

```
recursive clients: 0/0/1000
tcp clients: 0/100
server is up and running
alfa:~#
```

Configuración de los clientes

Comandos y archivos útiles

- /etc/resolv.conf
- vi

Operaciones

1. Comprobar la configuración del servidor alfa para que se consulte a sí mismo para todas las peticiones DNS.
2. Modificar la configuración de la estación de trabajo Ubuntu para que utilice el servidor alfa para todas las peticiones DNS.

Archivos modificados

Archivo /etc/resolv.conf en el servidor alfa

Archivo

```
nameserver    192.168.200.101
```

/etc/resolv.conf en la estación de trabajo

2.

```
nameserver    192.168.200.101
```

Configuración del servidor de caché

Si la instalación ha tenido éxito, el servidor conoce la ruta hacia el exterior y el puerto 53 no está filtrado, su servidor debe poder realizar cualquier resolución en el espacio de nombres público.

a. Comprobación de la resolución de nombres en el espacio de nombres público

Comandos útiles

- ping

Operaciones

1. Hacer un ping en una dirección pública conocida.

Resumen de los comandos y resultado por pantalla

Ping en una dirección pública:

Cabe

```
alfa:~# ping www.gnu.org
PING gnu.org (199.232.41.10) 56(84) bytes of data.
64 bytes from www.gnu.org (199.232.41.10): icmp_seq=1 ttl=52 time=136 ms
64 bytes from www.gnu.org (199.232.41.10): icmp_seq=2 ttl=52 time=140 ms
alfa:~#
```

recordar que cuando se comprueba una resolución con el comando ping, es la primera línea la que interesa, la

que traduce el nombre en dirección IP y no la posible respuesta de ping, que se puede filtrar.

b. Pruebas de puesta en caché (opcionales)

Comandos útiles

- ping

Operaciones

1. Justo después de una resolución con éxito, hacer que el servidor no pueda salir de la red desconectando el router de su switch, por ejemplo.
2. Volver a hacer lo mismo mientras que el servidor no acceda al exterior.
3. Hacer un ping en otro nombre cualquiera.

Resumen de los comandos y resultado por pantalla

Ping con un valor conservado en caché:

```
alfa:~# ping www.gnu.org
PING gnu.org (199.232.41.10) 56(84) bytes of data.
alfa:~#
```

Ping con un valor nuevo:

```
alfa:~# ping www.kernel.org
ping unknown host www.kernel.org
alfa:~#
```

c. Redirección

Ahora dispone de un servidor capaz de hacer resoluciones de nombres. Para que su servidor pueda beneficiarse de la caché del servidor de su proveedor de acceso, declare una redirección.

Comandos y archivos útiles

- named.conf (named.conf.options)
- vi

Operaciones

1. Editar el archivo /etc/bind/named.conf.options.
2. Descomentar la línea de forwarders suprimiendo la doble barra así como las dos líneas inmediatamente siguientes.
3. Reemplazar 0.0.0.0 por la dirección IP del servidor DNS de su ISP.
4. Reiniciar el servicio mediante un comando a su elección.

Resumen de los comandos y resultado por pantalla

Archivo /etc/bind/named.conf.options:

Reinicio del

```
options {
        directory "/var/cache/bind";

        // If there is a firewall between you and nameservers you want
        // to talk to, you may need to fix the firewall to allow multiple
```

```
        // ports to talk.  See http://www.kb.cert.org/vuls/id/800113

        // If your ISP provided one or more IP addresses for stable
        // nameservers, you probably want to use them as forwarders.
        // Uncomment the following block, and insert the addresses replacing
        // the all-0's placeholder.

        forwarders {
                8.8.8.8;
        };

        auth-nxdomain no;    # conform to RFC1035
        listen-on-v6 { any; };
};
```

servicio:

3.

```
alfa:/etc/bind# /etc/init.d/bind9 restart
Stopping domain name service...: bind9.
Starting domain name service...: bind9.
alfa:/etc/bind#
```

Creación de zonas personalizadas directas e inversas

Animado por el éxito en estas tareas, decide utilizar su servidor DNS para albergar una zona local. Esta zona le permitirá crear registros que hagan referencia a sus recursos locales como por ejemplo impresoras en red. Para seguir la metodología, decide crear la zona directa pas.net y una zona inversa correspondiente a la red 192.168.200.0.

a. Creación de archivos de zonas en el servidor alfa

Decide crear los archivos para la zona directa pas.net y para la zona inversa 200.168.192.in-addr.arpa. Estos archivos deberán declarar alfa como servidor maestro para estas zonas. La dirección de email del contacto administrativo es root@pas.net.

Comandos y archivos útiles

- archivos de zona
- vi

Operaciones

1. En /etc/bind, crear dos archivos: db.pas.net y db.192.168.200.
2. En ambos archivos, crear el registro SOA con alfa como servidor autoritario. Hay que basarse en /etc/bind/db.empty para conocer los valores de caché por defecto. No hay que olvidar que las zonas son FQDN y por lo tanto deben terminar con un punto. La dirección de correo electrónico del contacto administrativo es root@pas.net.
3. Crear registros NS para cada uno de los archivos. Para ambas zonas, el servidor alfa es el servidor de nombres.
4. Crear en la zona directa un registro de tipo A para asociar una dirección IP al servidor alfa.

Resumen de los comandos y resultado por pantalla

Archivo /etc/bind/db.pas.net:

Archivo

```
$TTL    86400
pas.net.    IN  SOA      alfa.pas.net. root.pas.net. (
```

```
        1
        604800
        86400
        2419200
        86400 )

pas.net.   IN   NS      alfa.pas.net.
alfa.pas.net.    IN    A     192.168.200.101
```

/etc/bind/db.192.168.200:

b.

```
$TTL    86400
200.168.192.in-addr.arpa.   IN  SOA     alfa.pas.net. root.pas.net. (
        1
        604800
        86400
        2419200
        86400 )

200.168.192.in-addr.arpa.    IN   NS    alfa.pas.net.
```

Declaración de archivos de zona

Ahora, el objetivo es hacer saber al servidor que debe cargar ambos archivos de zona que se acaban de crear en cada inicio del servicio.

Comandos y archivos útiles

- named.conf (named.conf.local)
- vi

Operaciones

1. En /etc/bind/named.conf.local, crear una sección zone que haga referencia a su zona directa como maestro de zona.
2. En /etc/bind/named.conf.local, crear una sección zone que haga referencia a su zona inversa como maestro de zona.
3. Recargue el servicio.

Resumen de los comandos y resultado por pantalla

Archivo /etc/bind/named.conf.local:

```
//
// Do any local configuration here
//

// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";
zone "pas.net" {
        type master;
        file "/etc/bind/db.pas.net";
        };

zone "200.168.192.in-addr.arpa" {
        type master;
        file "/etc/bind/db.192.168.200";
        };
```

Recarga del servicio:

```
alfa:/etc/bind# /etc/init.d/bind9 restart
Stopping domain name service...: bind9.
Starting domain name service...: bind9.
alfa:/etc/bind#
```

Es imprescindible que el servicio se reinicie sin problemas. Un **tail** de **/var/log/syslog** servirá para saber si las zonas se han cargado con éxito y si el servicio se ha iniciado correctamente.

c. Creación de registros

Ahora decide crear algunos registros de distinta naturaleza para comprobar el funcionamiento de su servidor.

Comandos y archivos útiles

- Archivos de zona
- vi

Operaciones

1. En su archivo de zona directa, cree el registro **beta.pas.net** de tipo **A**correspondiente a la dirección IP del servidor beta.
2. En su archivo de zona directa, cree el registro **servidor-a** de tipo **CNAME**correspondiente al FQDN **alfa.pas.net**.
3. En su archivo de zona inversa, cree el registro **101** de tipo **PTR** correspondiente al nombre del servidor alfa.
4. En su archivo de zona inversa, cree el registro **102** de tipo **PTR** correspondiente al nombre del servidor beta.
5. Recargue la información de la zona.

Resumen de los comandos y resultado por pantalla

Registros de recursos en el archivo de zona db.pas.net:

```
beta.pas.net. IN   A       192.168.200.102
servidor-a    IN   CNAME   alfa.pas.net.
```

Registros de recursos en el archivo de zona 200.168.192.in-addr.arpa:

```
101 IN PTR alfa.pas.net.
102 IN PTR beta.pas.net.
```

Recarga de la información de la zona:

```
alfa:/etc/bind# rndc reload
server reload successful
alfa:/etc/bind#
```

4.

Consultas al servidor

Estando decidio a comprobar que todo funciona correctamente, realiza algunas pruebas desde la estación de trabajo Ubuntu.

a. Utilización de nslookup

Comando útil

- nslookup

Operaciones

1. Especifique que el servidor al que se realizan las consultas es alfa.
2. Solicite la dirección correspondiente al nombre **alfa.pas.net**.
3. Solicite la dirección correspondiente al nombre **servidor-a.pas.net**.
4. Solicite el nombre correspondiente a la dirección **192.168.200.102**.

Resumen de los comandos y resultado por pantalla

Resolución desde la estación con nslookup

b.

```
usuario@estacion:~$ nslookup
> server 192.168.200.101
Default server: 192.168.200.101
Address: 192.168.200.101#53
> alfa.pas.net
Server:  192.168.200.101
Address: 192.168.200.101#53

Name: alfa.pas.net
Address: 192.168.200.101
> servidor-a.pas.net
Server:  192.168.200.101
Address: 192.168.200.101#53

servidor-a.pas.net canonical name = alfa.pas.net.
Name: alfa.pas.net
Address: 192.168.200.101
> 192.168.200.102
Server:  192.168.200.101
Address: 192.168.200.101#53

102.200.168.192.in-addr.arpa name = beta.pas.net.
>
```

Utilización de dig

Comando útil

- dig

Operaciones

1. Solicite al servidor alfa la dirección de beta.pas.net.

Resumen de los comandos y resultado por pantalla

5.

```
usuario@estacion:~$ dig @192.168.200.101 beta.pas.net
```

```
; <<>> DiG 9.6.1-P1 <<>> @192.168.200.101 beta.pas.net
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 34012
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1

;; QUESTION SECTION:
;beta.pas.net.                  IN      A

;; ANSWER SECTION:
beta.pas.net.           86400   IN      A       192.168.200.102

;; AUTHORITY SECTION:
pas.net.    86400   IN      NS      alfa.pas.net.

;; ADDITIONAL SECTION:
alfa.pas.net.       86400   IN      A       192.168.200.101

;; Query time: 10 msec
;; SERVER: 192.168.200.101#53(192.168.200.101)
;; WHEN: Tue Jul 26 19:49:25 2011
;; MSG SIZE  rcvd: 82

usuario@estacion:~$
```

Creación de un servidor secundario

Inquieto por la disponibilidad del servicio, decide replicar sus zonas en un segundo servidor. Por tanto, decide usar el servidor beta como servidor secundario para sus zonas.

Las distribuciones Red Hat ofrecen un servicio bind con el chroot realizado por defecto. Sin ningún tipo de problemas, el entorno de trabajo se situará de forma integral en el directorio **/var/named/chroot**y el funcionamiento con el chroot realizado estará ya configurado en el script de inicio del servicio.

a. Configuración del servidor secundario

Durante la sincronización, los archivos de zona se crearán de manera local en el servidor beta. Hay que asegurar que la cuenta de servicio (named) tiene permisos de escritura en su directorio de almacenamiento.

Comandos y archivos útiles

- named.conf
- chmod

Operaciones

1. Crear el archivo **named.conf** en el directorio **/var/named/chroot/etc**.
2. Declarar las dos zonas esclavas **pas.net** y **200.168.192.in-addr.arpa** que tienen como maestro el servidor alfa. Asegurarse de que los archivos de zona se almacenan en el directorio del chroot **/var/named**.
3. Hacer que el grupo de la cuenta de servicio **named** pueda escribir en el directorio de almacenamiento de los archivos de zona (/var/named/chroot/var/named).
4. Iniciar el servicio y comprobar que no hay errores.

Resumen de los comandos y resultado por pantalla

Archivo /var/named/chroot/etc/named.conf :

```
zone "pas.net" {
        type slave;
        masters { 192.168.200.101 ; };
```

```
        file "/var/named/pas.net";
        };
zone "200.168.192.in-addr.arpa" {
        type slave;
        masters { 192.168.200.101 ; };
        file "/var/named/200.168.192.in-addr.arpa";
        };
```

Asignación de permisos a la cuenta de servicio named:

```
[root@beta var]# ls -ld /var/named/chroot/var/named/
drwxr-x--- 4 root named 4096 jul 26 20:01 /var/named/chroot/var/named/
[root@beta var]# chmod g+w /var/named/chroot/var/named/
[root@beta var]# ls -ld /var/named/chroot/var/named/
drwxrwx--- 4 root named 4096 jul 26 20:01 /var/named/chroot/var/named/
[root@beta var]#
```

Inicio del servicio:

```
[root@beta var]# service named start
Iniciando de named:
[  OK  ]
[root@beta var]#
```

Comprobación:

b.

```
[root@beta var]# tail /var/log/messages
Jul 26 20:02:48 beta named[641]: running
Jul 26 20:02:48 beta named[641]: zone pas.net/IN: Transfer started.
Jul 26 20:02:48 beta named[641]: transfer of 'pas.net/IN' from 192.168.200.101#53:
connected using 192.168.200.51#58348
Jul 26 20:02:48 beta named[641]: zone pas.net/IN: transferred serial 1
Jul 26 20:02:48 beta named[641]: transfer of 'pas.net/IN' from 192.168.200.101#53:
end of transfer
Jul 26 20:02:49 beta named[641]: zone 200.168.192.in-addr.arpa/IN: Transfer started.
Jul 26 20:02:49 beta named[641]: transfer of '200.168.192.in-addr.arpa/IN' from
192.168.200.101#53: connected using 192.168.200.51#48705
Jul 26 20:02:49 beta named[641]: zone 200.168.192.in-addr.arpa/IN: transferred serial 1
Jul 26 20:02:49 beta named[641]: transfer of '200.168.192.in-addr.arpa/IN' from
192.168.200.101#53: end of transfer
[root@beta var]#
[root@beta var]# ls /var/named/chroot/var/named/
200.168.192.in-addr.arpa  data  pas.net  slaves
[root@beta var]#
```

Configuración del servidor primario

Sólo nos falta indicar al servidor maestro que ahora debe trabajar con un compañero.

Comandos y archivos útiles

- Archivos de zona
- dig
- tail
- vi

Operaciones

1. Declarar el servidor beta como nuevo servidor de tipo NS para sus dos zonas.
2. Añadir un registro de tipo A que haga referencia la estación de trabajo Ubuntu.
3. Incrementar el número de serie de las zonas.
4. Recargar las zonas.
5. Comprobar consultando el registro de sistema del servidor alfa que el reinicio del servicio se ha realizado correctamente.
6. Comprobar en el servidor beta que las modificaciones se han transmitido correctamente.

Resumen de los comandos y resultado por pantalla

Archivo de zona directa modificado:

```
$TTL    86400
pas.net.        IN      SOA     alfa.pas.net. root.pas.net. (
                              2         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                          86400 )       ; Negative Cache TTL
;
pas.net.        IN      NS      alfa.pas.net.
pas.net.        IN      NS      beta.pas.net.
alfa.pas.net.   IN      A       192.168.200.101
beta.pas.net.   IN      A       192.168.200.102
servidor-a      IN      CNAME   alfa.pas.net.
alpha           IN      CNAME   alfa
estacion        IN      A       192.168.200.199
```

Archivo de zona inversa modificado:

```
$TTL    86400
200.168.192.in-addr.arpa.       IN      SOA     alfa.pas.net. root.pas.net. (
                              2         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                          86400 )       ; Negative Cache TTL
;
200.168.192.in-addr.arpa.       IN      NS      alfa.pas.net.
200.168.192.in-addr.arpa.       IN      NS      beta.pas.net.
101             IN      PTR     alfa.pas.net.
102             IN      PTR     beta.pas.net.
199             IN      PTR     estacion.pas.net.
```

Recarga de las zonas:

```
alfa:/etc/bind# rndc reload
server reload successful
alfa:/etc/bind# tail /var/log/daemon.log
Jul 26 20:29:56 alpha named[4638]: loading configuration from '/etc/bind/named.conf'
Jul 26 20:29:56 alpha named[4638]: using default UDP/IPv4 port range: [1024, 65535]
Jul 26 20:29:56 alpha named[4638]: using default UDP/IPv6 port range: [1024, 65535]
Jul 26 20:29:56 alpha named[4638]: reloading configuration succeeded
Jul 26 20:29:56 alpha named[4638]: reloading zones succeeded
Jul 26 20:29:56 alpha named[4638]: zone 200.168.192.in-addr.arpa/IN: loaded serial 2
Jul 26 20:29:56 alpha named[4638]: zone 200.168.192.in-addr.arpa/IN: sending
```

```
notifies (serial 2)
Jul 26 20:29:56 alpha named[4638]: zone pas.net/IN: loaded serial 2
Jul 26 20:29:56 alpha named[4638]: zone pas.net/IN: sending notifies (serial 2)
alfa:/etc/bind#
```

Comprobación en el servidor esclavo con dig:

```
[root@beta named]# dig @127.0.0.1 estacion.pas.net

; <<>> DiG 9.3.6-P1-RedHat-9.3.6-4.P1.el5_4.2 <<>> @127.0.0.1 estacion.pas.net
; (1 server found)
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 27701
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 2

;; QUESTION SECTION:
;estacion.pas.net.                      IN      A

;; ANSWER SECTION:
estacion.pas.net.       86400   IN      A       192.168.200.199

;; AUTHORITY SECTION:
pas.net.                86400   IN      NS      beta.pas.net.
pas.net.                86400   IN      NS      alfa.pas.net.

;; ADDITIONAL SECTION:
beta.pas.net.           86400   IN      A       192.168.200.102
alfa.pas.net.           86400   IN      A       192.168.200.101

;; Query time: 5 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Fri Jul 26 20:31:37 2011
;; MSG SIZE  rcvd: 119

[root@beta named]#
```

Consulta de los registros en el servidor esclavo:

```
[root@beta named]# tail /var/log/messages
Jul 26 20:30:54 beta named[1352]: client 192.168.200.102#58123:
received notify for zone 'pas.net'
Jul 26 20:30:54 beta named[1352]: zone pas.net/IN: refused notify
from non-master: 192.168.200.102#58123
Jul 26 20:30:58 beta named[1352]: client 192.168.200.101#26310:
received notify for zone '200.168.192.in-addr.arpa'
Jul 26 20:30:58 beta named[1352]: zone 200.168.192.in-addr.arpa/IN: Transfer started.
Jul 26 20:30:59 beta named[1352]: transfer of '200.168.192.in-
addr.arpa/IN' from 192.168.200.101#53: connected using 192.168.200.102#55607
Jul 26 20:30:59 beta named[1352]: zone 200.168.192.in-addr.arpa/IN: transferred serial 2
Jul 26 20:30:59 beta named[1352]: transfer of '200.168.192.in-
addr.arpa/IN' from 192.168.200.101#53: end of transfer
Jul 26 20:30:59 beta named[1352]: zone 200.168.192.in-addr.arpa/IN:
sending notifies (serial 2)
Jul 26 20:30:59 beta named[1352]: client 192.168.200.102#62475:
received notify for zone '200.168.192.in-addr.arpa'
Jul 26 20:30:59 beta named[1352]: zone 200.168.192.in-addr.arpa/IN:
refused notify from non-master: 192.168.200.102#62475
[root@beta named]#
```

Requisitos y objetivos

1. Requisitos

Los conocimientos adquiridos en la certificación LPI nivel 1, especialmente:

- ☒ Edición de archivos de texto.

2. Objetivos

Al final del capítulo, será capaz de:

- ☒ Configurar un servidor Apache básico.
- ☒ Conocer las principales directivas de Apache.
- ☒ Conocer los principales módulos de Apache.
- ☒ Configurar el acceso de los usuarios a sus páginas personales.
- ☒ Configurar hosts virtuales.
- ☒ Configurar la autentificación de usuarios.
- ☒ Conocer los conceptos de certificados digitales.
- ☒ Configurar un sitio web seguro con SSL.
- ☒ Conocer el funcionamiento de un servidor proxy.
- ☒ Asegurar una configuración básica del servidor proxy squid.

Configuración básica de un servidor Apache

1. Apache y los servidores web

El famoso servidor Web Apache es el más conocido y por lo menos desde 1996 el más extendido en Internet. Su popularidad se debe a su gran estabilidad y a su buena tolerancia a la carga. Está disponible en todas las arquitecturas de servidor de contenido dinámico LAMP y WAMP (Linux/Windows-Apache-MySQL-PHP) y su estructura modular lo convierte en un buen candidato para la mayoría de las situaciones.

El servidor web se compone de un script de inicio del servicio que, en función de un archivo de configuración, cargará los daemons apache y los posibles módulos de funcionamiento.

La riqueza funcional de Apache implica a menudo un archivo de configuración impresionante. Sin embargo, la configuración de un servidor básico es bastante sencilla de realizar.

2. Archivo de configuración

a. Formato del archivo de configuración

El archivo de configuración de Apache, que según la versión y las opciones de compilación puede ser **httpd.conf**, **apache.conf** o **apache2.conf**, se compone de directivas seguidas de valores. Algunas directivas se integran en contenedores para limitar su ámbito de aplicación.

Formato estándar del archivo de configuración de Apache

```
directiva1 valor1
directiva2 valor2
...
<directiva_de_contenedor valor>
  directiva3 valor3
  directiva4 valor4
  ...
</directiva_de_contenedor>
```

Como se puede ver, hay directivas escritas directamente y otras integradas en una sección en el archivo de configuración. Esta sección limita el ámbito de aplicación de las directivas a un determinado contexto de funcionamiento. La configuración de Apache consiste en elegir las directivas adecuadas, asignarles los valores adecuados y construir las secciones necesarias.

De entre todas las innumerables directivas de Apache, algunas son fundamentales y deberán encontrarse en cualquier configuración de Apache.

Archivo de configuración: directivas comunes	
ServerRoot	Indica el directorio raíz de los archivos de configuración.
User	Determina la cuenta de usuario del proceso Apache.
Group	Determina el grupo de servicio propietario de los procesos Apache.
ErrorLog	Archivo de registro de errores.
Include	Indica un archivo de configuración anexo que se integrará en el archivo apache2.conf.
Listen	Indica el puerto en el que escucha el servidor.
DocumentRoot	Indica el directorio que contiene los archivos html.

Ejemplo de archivo de configuración mínimo

Aunque esta configuración pueda funcionar, el servidor carecerá de cierta comodidad de uso. Por ejemplo, no reconocerá los archivos index.html o index.htm. Por lo tanto se deberá proporcionar en la URL el nombre de los archivos html deseados tecleando por ejemplo http://A.B.C.D/index.html si el directorio /var/www contiene el

*archivo index.html. Además, los scripts de inicio del servicio se podrían confundir por este archivo de configuración mínimo que está estructurado en una sola pieza. Sin lugar a dudas, es recomendable iniciar el ejecutable **apache** directamente sin usar el script.*

```
ServerRoot /etc/apache2
User www-data
Group www-data
ErrorLog /var/log/apache2/error.log
Listen 80
DocumentRoot /var/www
```

b. Directivas de contenedor

Ya se ha mostrado que las directivas sirven para aplicar un elemento de configuración al servidor. Por ejemplo, la línea de configuración **Listen 80** en el archivo de configuración se compone de la directiva **Listen**, que indica en qué puerto el servidor debe esperar peticiones, y del valor 80 que es el puerto HTTP estándar. Ubicada directamente en el archivo de configuración, esta directiva se aplicará en todo el servidor.

Sin embargo, hay elementos de configuración que sólo afectan a un aspecto funcional del servidor. Por ejemplo, hay directivas que sólo deberían aplicarse a una parte limitada de un sitio web, como páginas web protegidas que estén situadas en una estructura de carpetas específica del sistema de archivos. Para este tipo de usos, Apache utiliza las directivas de contenedor.

Las directivas de contenedor tienen dos objetivos: agrupar un conjunto de directivas de configuración y aplicarlas a una parte limitada del servidor Apache.

Sintaxis genérica de una directiva de contenedor

```
<directiva_de_contenedor valor>
  directiva3 valor3
  directiva4 valor4
  ...
</directiva_de_contenedor>
```

Observe que cualquier sección definida por una directiva de contenedor rodea la directiva con los caracteres **<** **>** y que la sección termina con el nombre de la directiva precedido con una barra. Entre el inicio y el final de la sección se encuentran todas las directivas normales que se aplicarán en el contexto funcional definido por la directiva de contenedor.

Ejemplo de directiva de contenedor

La directiva Directory es sin duda la directiva de contenedor más fácil de aprender. Admite como argumento un directorio y define parámetros específicos del contenido web situado en este directorio. En el ejemplo mostrado a continuación, se indica que en el directorio /var/www/especial, y sólo en este directorio, el servidor puede seguir los enlaces simbólicos durante la lectura de páginas web.

```
<Directory /var/www/especial>
  Options FollowSymLinks
</Directory>
```

c. Validación de la sintaxis

Es posible (y prudente) validar la sintaxis de un archivo de configuración antes de iniciar el servicio.

Validación de la sintaxis del archivo apache2.conf

```
ejec_apache -t
```

Donde *ejec_apache* representa el ejecutable de inicio del servidor Apache. Los valores más frecuentes son **httpd**, **apache** y **apache2**.

d. Inicio y parada del servidor

En un entorno de producción, el inicio y la parada del servidor se realizarán mediante el uso del script de gestión del servicio ubicado en /etc/init.d. Sin embargo, en fase de pruebas, es útil iniciarlo y pararlo manualmente.

Inicio manual del servidor Apache

```
ejec_apache -k start
```

Parada del servidor Apache

```
ejec_apache -k stop
```

Donde *ejec_apache* representa el ejecutable de inicio del servidor Apache. Los valores habituales son **httpd**, **apache** y **apache2**.

También se puede controlar el daemon apache mediante el comando de control apache2ctl con el uso, entre otros, de los mismos parámetros start y stop.

3. Módulos Apache

a. Carga de módulos

El servidor web Apache tiene una estructura modular, es decir, las funciones fundamentales del servidor se proporcionan con el ejecutable apache, mientras que las funciones adicionales o suplementarias se añaden a través de módulos cargados bajo demanda desde el archivo de configuración. Estos módulos requieren a menudo elementos de configuración adicionales. Las directivas asociadas a los módulos deben añadirse también al archivo de configuración.

Formato estándar de carga de directivas en apache2.conf

```
LoadModule id_módulo archivo_módulo
directiva valor
```

Archivo de configuración: Carga de módulos	
LoadModule	Directiva de carga de módulos.
id_módulo	Identificador del módulo. Valor estándar asociado a cada módulo.
archivo_módulo	La ruta absoluta del archivo de módulo proporcionado con Apache.

Ejemplo de carga de un módulo

*En este ejemplo, el módulo cargado es **dir_module** cuya función es simplificar la escritura de las URL por los usuarios y mostrar un archivo html (en general index.html) incluso si éste no se ha facilitado. El archivo ejecutable de este módulo es **mod_dir.so**. Una vez que se ha cargado el módulo, se puede llamar a la directiva **DirectoryIndex** que solicita la carga de index.html si no hay ningún archivo en la URL.*

```
LoadModule dir_module /usr/lib/apache2/modules/mod_dir.so
DirectoryIndex index.html
```

b. Visualización de módulos

El comando apache, cuando se utiliza de forma interactiva, permite visualizar los módulos cargados. Los módulos pueden tener dos orígenes: se han llamado con el comando load desde el archivo de configuración o se han compilado con el núcleo del programa y se cargan automáticamente.

Visualización de módulos compilados en el programa

```
ejec_apache -l
```

Visualización de módulos cargados

```
ejec_apache -M
```

Ejemplo de visualización de módulos cargados

La opción -M muestra los módulos estáticos y los módulos cargados desde el archivo de configuración con la directiva LoadModule.

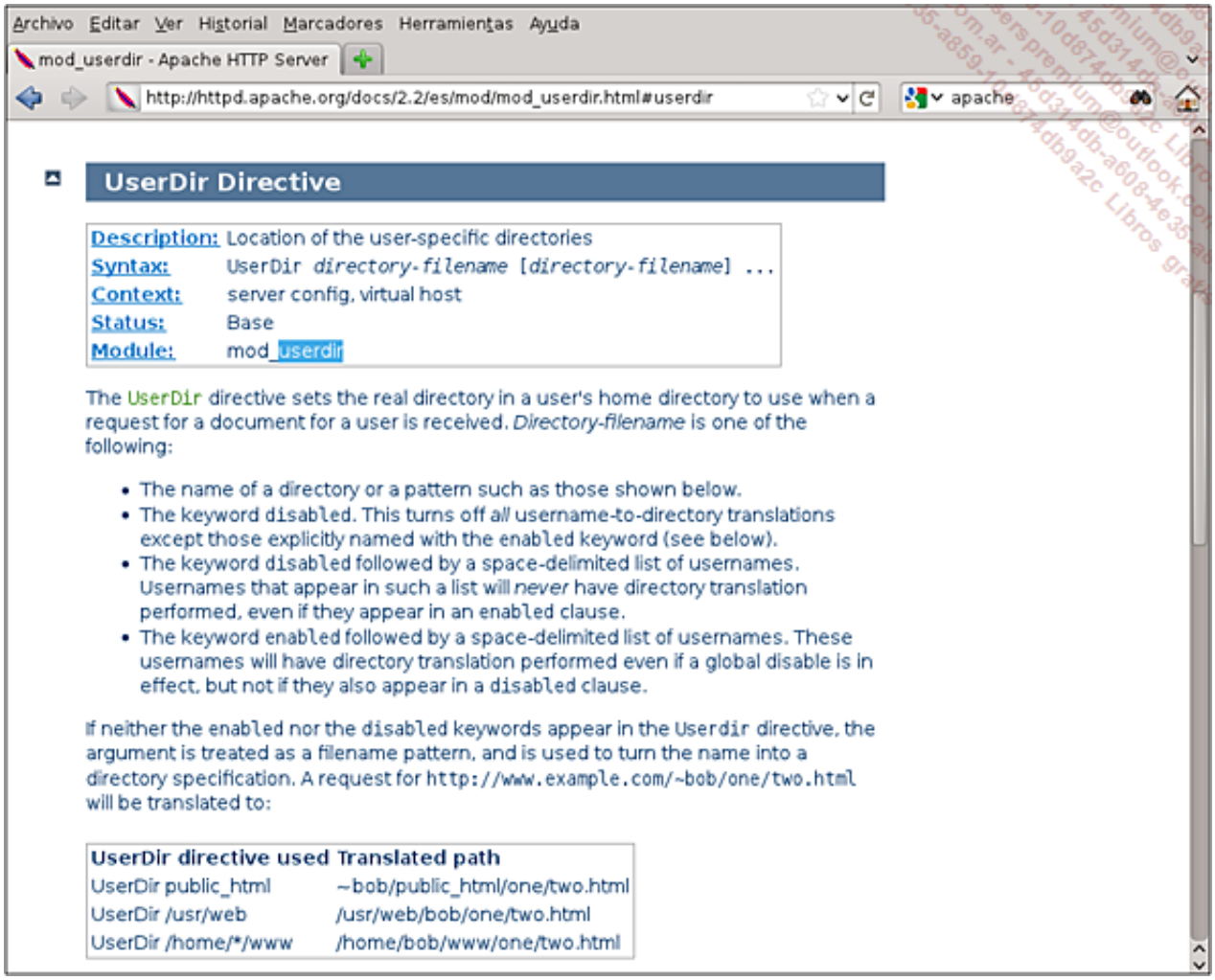
```
alfa:/etc/apache2# apache2 -l
Compiled in modules:
  core.c
  mod_log_config.c
  mod_logio.c
  worker.c
  http_core.c
  mod_so.c
alfa:/etc/apache2# apache2 -M
Loaded Modules:
 core_module (static)
 log_config_module (static)
 logio_module (static)
 mpm_worker_module (static)
 http_module (static)
 so_module (static)
 alias_module (shared)
 auth_basic_module (shared)
 authn_file_module (shared)
 authz_default_module (shared)
 authz_groupfile_module (shared)
 authz_host_module (shared)
 authz_user_module (shared)
 autoindex_module (shared)
 cgid_module (shared)
 deflate_module (shared)
 dir_module (shared)
 env_module (shared)
 mime_module (shared)
 negotiation_module (shared)
 setenvif_module (shared)
 status_module (shared)
Syntax OK
alfa:/etc/apache2#
```

c. Elección de los módulos

Los módulos dependen naturalmente del uso que se haga del servidor. Para determinar el uso que se le va a dar, hay que identificar las necesidades, determinar las directivas asociadas a estas necesidades y comprobar en la documentación en línea de Apache (sección Directivas para configurar la ejecución) cuáles son los módulos implicados.

Supongamos que se desea dar a los usuarios acceso a páginas personales situadas en su directorio personal. La directiva **UserDir** sirve precisamente para este fin. Admite como parámetro una ruta relativa que indica la ubicación del directorio personal del usuario donde se colocarán los recursos html del mismo. Los documentos por lo tanto estarán accesibles mediante la url http://servidor/~usuario/.

Si se consulta esta directiva en la documentación en línea, se puede comprobar que depende del módulo **mod_userdir**. Por tanto, habrá que configurar esta directiva y asegurarse de que se carga el módulo.



Ejemplo de configuración del acceso a la carpeta de usuarios

Con la configuración del acceso a la carpeta de usuarios siguiente, la url http://servidor/~usuario/doc.html dará acceso al archivo /home/usuario/web/doc.html en el servidor.

```
LoadModule userdir_module modules/mod_userdir.so
UserDir web
```

4. Gestión de recursos

Contrariamente a una creencia popular, el servidor Apache no es necesariamente el servidor más rápido del mercado y productos con código más simple ofrecen tiempos de respuesta más rápidos con hardware equivalente. Sin embargo, Apache, con su gestión inteligente de recursos y la prealocación de procesos, ofrece una mejor escalabilidad respecto a la carga.

Observemos los procesos iniciados por Apache en un servidor inactivo:

```
alfa:~# ps -ef | grep apache
root      3244     1  0 Feb05 ?        00:00:04 /usr/sbin/apache2 -k start
www-data  3245  3244  0 Feb05 ?        00:00:00 /usr/sbin/apache2 -k start
www-data  3247  3244  0 Feb05 ?        00:00:00 /usr/sbin/apache2 -k start
www-data  3252  3244  0 Feb05 ?        00:00:00 /usr/sbin/apache2 -k start
alfa:~#
```

Se puede comprobar la existencia de un proceso ejecutado por root y de otros tres por la cuenta de servicio www-data. El primer proceso sólo sirve para iniciar los demás, que procesan las peticiones de los clientes. El servidor mostrado en este ejemplo no gestiona ninguna conexión cliente y, sin embargo, hay tres procesos previamente creados listos para gestionar toda la demanda de los clientes. La gestión del primer servicio por la cuenta root es obligatoria, ya que es el único con privilegios para abrir el puerto 80 en un sistema Linux.

Hosts virtuales

Es frecuente que un servidor Apache físico albergue varios sitios web distintos. Es lo que permite a las empresas de alojamiento gestionar sitios web de decenas de clientes en un solo servidor. Esta tecnología se conoce en el mundo Apache con el nombre de "Virtual Host" (host virtual).

1. Configuración global

a. Gestión de contenidos

Si un servidor tiene que gestionar hosts virtuales, es porque debe albergar varios contenidos o sitios web distintos. Simplemente será necesario albergar cada uno de estos contenidos en directorios diferentes y debidamente identificados. El alojamiento de un gran número de sitios web en un solo servidor requiere cierta disciplina y convenios de nomenclatura estrictos.

b. Organización de sitios virtuales

El archivo de configuración debe modificarse ligeramente. Cada host virtual lee sus elementos de configuración específicos en contenedores declarados mediante la directiva **VirtualHost**. Algunas directivas de tipo general (DocumentRoot, por ejemplo) dejarán de usarse y deberán especificarse en cada uno de los hosts virtuales en el contenedor correspondiente.

Si un servidor Apache gestiona hosts virtuales, sólo hay que hacer este cambio. Es decir, no hay configuración estándar a la que se añadan configuraciones específicas para los hosts virtuales. Una vez que se configuran los hosts virtuales, todo acceso al servidor se realiza por un host virtual, incluso si es el sitio básico.

2. Configuración de hosts virtuales

Hay dos técnicas de implementación de hosts virtuales: los hosts virtuales por dirección IP -donde el servidor proporciona contenido distinto según la IP con la que se contacta con él- y los hosts virtuales por nombre de host -donde el servidor proporciona contenido distinto en función del nombre de host presente en la URL con la que se contacta con él-.

a. Hosts virtuales por dirección IP

Aunque el servidor soporta hosts virtuales por dirección IP, raramente se usa. En esta configuración, el servidor dispone de varias direcciones IP y responde de forma distinta según la interfaz a la que llegue cada petición HTTP. Se tiene que crear una directiva **VirtualHost** para cada dirección IP que esté en uso. Esta directiva contendrá la declaración del directorio que contenga los datos html correspondientes al sitio virtual. Esta declaración se realiza mediante la directiva **DocumentRoot**.

Declaración en el archivo de configuración Apache

```
<VirtualHost dirección_1:80>
ServerName nombre1
DocumentRoot dir1
</VirtualHost>

<VirtualHost dirección_2:80>
ServerName nombre2
DocumentRoot dir2
</VirtualHost>
```

Archivo de configuración: declaración de hosts virtuales	
<VirtualHost>	Declaración de un host virtual: todas las directivas que contiene afectarán al host virtual.
dirección_x:80	El host virtual se elegirá si la petición que se recibe tiene como dirección IP destino esta dirección del servidor configurada en

	el puerto 80.
ServerName*nombre*	Opcional si existe una resolución de nombres inversa. Si no, puede devolver un mensaje de error no bloqueante.
DocumentRoot*dir*	Las páginas web de este host virtual están en el directorio *dir*.

b. Hosts virtuales por nombre de host

El soporte de hosts virtuales por nombre de host requiere que se use la directiva**NameVirtualHost** que debe ubicarse en el contexto general del archivo de configuración y tantas directivas de contenedor **VirtualHost** como sitios virtuales albergue el servidor. Para entender adecuadamente la organización de los sitios virtuales, hay que saber que no hay un sitio principal y varios sitios virtuales sino que todo sitio alojado en el servidor es un sitio virtual.

Como el sitio virtual se reconoce a partir de su nombre de host, la directiva **ServerName** debe estar presente de forma sistemática en los contenedores de host virtual y el nombre asociado debe ser precisamente el que usarán los clientes para acceder al servidor (el nombre tiene que estar en la URL).

La última directiva obligatoria en una declaración de host virtual es la directiva **DocumentRoot** que determina la ubicación de los datos web asociados al sitio virtual.

Declaración del sitio virtual en el archivo de configuración Apache

```
NameVirtualHost *:80

<VirtualHost *:80>
ServerName nombre1
DocumentRoot dir1
</VirtualHost>

<VirtualHost *:80>
ServerName nombre2
DocumentRoot dir2
</VirtualHost>
```

Archivo de configuración: declaración de hosts virtuales	
NameVirtualHost *:80	Se gestionan los hosts virtuales por nombre. La escucha de las peticiones se realiza a través del puerto 80.
<VirtualHost>	Declaración de un host virtual: todo lo que se encuentre en el contenedor afectará al host virtual.
*:80	El host virtual se elegirá para las peticiones que se reciban en cualquier dirección IP del servidor por el puerto 80.
ServerName*nombre*	Este host virtual se elegirá para las peticiones que tengan el nombre de servidor *nombre*. Es decir, se activará para las peticiones a *http://nombre*.
DocumentRoot*dir*	Las páginas web de este host virtual se almacenarán en el directorio *dir*.

Restricción de acceso a usuarios

Apache ofrece múltiples posibilidades de restricción de acceso a usuarios y existen muchos medios para gestionar el acceso a datos. Cabe decir que este párrafo sólo trata los métodos de autentificación de los usuarios y no explica nada acerca de la confidencialidad. Es decir, que el acceso a los datos podrá someterse a la autentificación, pero las páginas web circularán sin encriptar entre el servidor y el navegador. Si se desea tener confidencialidad en la comunicación, se tendrá que usar el protocolo SSL que se verá más adelante.

1. Restricción de acceso a páginas web

a. Declaración del directorio que se desea proteger

La restricción se realiza en directorios cuyo contenido sólo se puede ver una vez el usuario se ha autentificado. Si se desea restringir el acceso a un sitio entero, basta con restringir el acceso al directorio raíz del contenido web. Sin embargo, en la práctica se desea tener una página de inicio disponible para todo el mundo y que toda navegación a partir de ésta esté sometida a la autentificación del usuario.

Declaración de una sección de directorio en apache2.conf

```
<Directory directorio>
...
</Directory>
```

Donde *directorio* representa la ruta absoluta del directorio al que hay que proteger el acceso. Cabe destacar los puntos suspensivos que representan las directivas específicas que se aplicarán al contenido de este directorio.

b. Directivas de autentificación

En la sección de directorio sujeta a la autentificación, hay que añadir el conjunto de directivas necesarias según el modo de autentificación que se escoja. Algunas de estas directivas se encontrarán en la mayoría de las circunstancias.

Solicitud de autentificación

```
AuthName "texto"
Require valid-user
Directiva_aut parámetro_aut
```

Archivo Apache2.conf: solicitud de autentificación para un directorio	
AuthName	Define el título del cuadro de diálogo que solicitará la autentificación del usuario.
texto	Título del cuadro de diálogo que solicitará la autentificación del usuario.
Require valid-user	Directiva que impone un funcionamiento específico con su parámetro valid-user que exige que el usuario esté correctamente autentificado.
Directiva_aut	La o las directivas de autentificación en función del método escogido.

Se ha empleado en este ejemplo el parámetro **valid-user** para establecer que cualquier usuario autentificado pueda acceder a los datos protegidos. Se podría haber usado **user x** o **group y** para limitar el acceso a un usuario o a los miembros de un grupo.

2. Autentificación local

a. Creación de una base de datos de cuentas locales

La creación de un archivo de cuentas locales para la autentificación de los visitantes apache es una forma sencilla y común de administrar usuarios. Se trata de un archivo único que contiene una línea por cada declaración de usuario. Por lo tanto, conviene usar este método en entornos con un número limitado de usuarios.

El comando **htpasswd** permite crear la base de datos de cuentas y trabajar con ella.

Creación de la primera cuenta de usuario

```
htpasswd -c archivo nombre_usuario
```

Añadir una cuenta de usuario

```
htpasswd archivo nombre_usuario
```

Eliminar una cuenta de usuario

```
htpasswd -D archivo nombre_usuario
```

Comando htpasswd: opciones y parámetros	
-c	Necesario si el archivo no existe todavía. Si el archivo ya existe, se sobreescribe.
archivo	El archivo que contiene la base de datos de cuentas.
nombre_usuario	El nombre de la cuenta creada.
-D	Elimina el usuario cuyo nombre se proporciona como parámetro.

Ejemplo de archivo de contraseñas

El archivo muestra en cada línea el nombre de la cuenta de usuario y su contraseña encriptada.

```
alfa:~# cat httpmdp
usuario:x4OpUoo9KBR1o
toto:o9zeMsxnhS45M
titi:hQcfUWksuIAmk
alfa:~#
```

b. Carga de módulos de autentificación

Algunos módulos tienen que cargarse para que se puedan reconocer las directivas. Habrá que cargar como mínimo el módulo **auth_basic** para permitir la autentificación mediante un archivo local, el módulo **authn_file** para gestionar esta autentificación y finalmente el módulo **authz_user**, que gestiona la autorización de acceso a las páginas protegidas. Esta profusión de módulos puede ser inquietante, pero un mínimo de rigor facilita las cosas: para cada una de las directivas empleadas, la documentación en línea especifica sistemáticamente qué módulos tienen que cargarse.

Carga de módulos

Los tres módulos son necesarios para la autentificación de usuarios.

```
LoadModule auth_basic_module /ruta/mod_auth_basic.so
LoadModule authn_file_module /ruta/mod_authn_file.so
LoadModule authz_user_module /ruta/mod_authz_user.so
```

c. Configuración de la autentificación local

En la sección de directorio sujeta a la autentificación, habrá que añadir a continuación las directivas necesarias para la autentificación local.

Directivas para la autentificación local

```
AuthType Basic
AuthUserFile archivo
```

Donde *archivo* representa el archivo que contiene la base de datos de cuentas de usuario usada para la autentificación con las contraseñas de los usuarios.

Ejemplo de archivo apache2.conf con autentificación

```
ServerRoot /etc/apache2
User www-data
Group www-data
ErrorLog /var/log/apache2/error.log
Listen 80
DocumentRoot /var/www

LoadModule dir_module /usr/lib/apache2/modules/mod_dir.so
DirectoryIndex index.html

LoadModule auth_basic_module /usr/lib/apache2/modules/mod_auth_basic.so
LoadModule authn_file_module /usr/lib/apache2/modules/mod_authn_file.so
LoadModule authz_user_module /usr/lib/apache2/modules/mod_authz_user.so
<Directory /var/www>
  AuthType basic
  AuthUserFile /root/httpmdp
  AuthName "Se necesita comprobar sus credenciales"
  Require valid-user
</Directory>
```

3. Autentificación mediante directorio LDAP

Existen varios métodos para usar un directorio LDAP como base de datos de autentificación. La configuración siguiente es uno de los posibles ejemplos.

a. Comprobación de la disponibilidad de la información del directorio

Supongamos que se dispone de un directorio LDAP con las siguientes características:

- Dirección IP: 192.168.1.11
- Contexto de cuentas de usuarios: ou=users,dc=direc,dc=es
- Nombre distinguido del administrador: cn=admin,dc=direc,dc=es

Ejemplo de consulta al directorio

Se recomienda encarecidamente comprobar que el directorio está adecuadamente en línea y accesible con los datos correctos (dirección IP y contexto LDAP) antes de configurar la autentificación LDAP para cualquier aplicación.

```
usuario@servidor# ldapsearch -x -D cn=admin,dc=direc,dc=es -W -h
192.168.1.11 -b ou=users,dc=direc,dc=es -s sub ObjectClass=*
Enter LDAP Password:
# extended LDIF
#
# LDAPv3
# base <dc=direc,dc=es> with scope subtree
# filter: (objectclass=*)
# requesting: ALL
#

# direc.es
dn: dc=direc,dc=es
objectClass: domain
dc: direc
(...)
```

```
# usuario, users.direc.es
dn: uid=usuario,ou=users,dc=direc,dc=es
objectClass: top
objectClass: posixAccount
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
(...)
usuario@servidor#
```

b. Carga de los módulos necesarios

Los módulos necesarios para la autentificación LDAP son los siguientes:

- auth_basic
- authn_file
- authz_user
- authnz_ldap

c. Configuración de la autentificación

Ahora se deberán utilizar las directivas de autentificación en una sección de directorio. El punto culminante de la configuración será la declaración de la directiva **AuthLDAPUrl**.

Directivas utilizadas para la autentificación LDAP

Aunque la sintaxis pueda parecer impresionante, no se debe al azar o a la imaginación de los desarrolladores. El formato URL LDAP se define en la RFC2255 y se utiliza en algunas aplicaciones.

```
AuthName "Texto"
AuthType Basic
Require Valid-user
AuthLDAPUrl ldap://192.168.1.11/ou=users,dc=direc,dc=es?cn?sub?objectclass=*
```

4. Autentificación simple mediante el archivo .htaccess

Las buenas prácticas de Apache recomiendan utilizar una directiva **Directory** cada vez que una configuración específica debe aplicarse al contenido de un directorio y ubicar en el bloque definido por esta directiva los elementos de configuración específicos a este directorio. Un método alternativo consiste en crear un archivo **.htaccess** en el directorio en cuestión e integrar en él las directivas que se deben aplicar en el contenido del directorio.

Aunque el mejor método es sin duda el uso de la directiva Directory, se recomienda (encarecidamente) para la certificación LPI estar familiarizado con el concepto de los archivos .htaccess.

Ejemplos comparativos de uso

Extracto del archivo de configuración Apache con la directiva Directory:

```
...
<Directory /var/www/prot>
AllowOverride all
authType basic
AuthName "Se necesita comprobar sus credenciales"
Require valid-user
AuthUserFile /etc/httpd/mdp
</Directory>
...
```

Contenido del archivo /var/www/prot/.htaccess utilizado para configurar una autentificación desde el directorio de

datos html:

```
authType basic
AuthName "Se necesita comprobar sus credenciales"
Require valid-user
AuthUserFile /etc/httpd/mdp
```

A veces, las dos configuraciones entran en conflicto, configurando por un lado el comportamiento específico del directorio en un contexto directory y, por otro lado, la misma directiva se configura de forma distinta en un archivo .htaccess del mismo directorio. En este caso, la directiva **AllowOverride**permite establecer qué configuración se utilizará.

Valores comunes de la directiva AllowOverride	
all	Valor por defecto: Cualquier directiva se autoriza en los archivos .htaccess.
none	Ninguna directiva se autoriza en los archivos .htaccess. Los archivos .htaccess serán ignorados.
AuthConfig	Autoriza las directivas relativas a los mecanismos de autentificación.

Se recomienda encarecidamente no aplicar la directiva AllowOverride All en el directorio raíz de su servidor web (¡valor por defecto!). Es mejor definir este valor a None, y crear una directiva Directory con la directiva AllowOverride configurada solamente para el directorio que nos interesa. Este método no sólo aumenta la seguridad, sino también el rendimiento evitando que el servidor web tenga que buscar un hipotético archivo .htaccess en cada uno de los directorios que se visitan.

Configuración de Apache con SSL

SSL (*Secure Socket Layer*) es un protocolo de seguridad de la capa de aplicación. Funciona con muchos protocolos, pero su uso más extendido es para asegurar http (https). SSL proporciona servicios de autentificación, de confidencialidad y de control de la integridad.

1. Criptografía y certificados

Explicar exhaustivamente qué son la criptografía y los certificados digitales X509 sobrepasarían por completo los objetivos de este libro, y su conocimiento detallado no es un requerimiento en la certificación LPI. Sin embargo, la utilización de certificados es necesaria para asegurar el acceso a las páginas web de un servidor mediante SSL (https).

a. Conceptos criptográficos

Cualquier infraestructura criptográfica se basa en algoritmos de encriptación. Estos algoritmos pueden pertenecer a tres familias distintas: los algoritmos simétricos donde se encripta y se desencripta gracias a una clave única, los algoritmos asimétricos, donde se dispone de un par de claves, una para encriptar y otra para desencriptar y, finalmente, los algoritmos de hash, de un solo sentido y que no usan la clave de cifrado.

La criptografía asimétrica usa dos claves. Por convenio, se decide que una de estas claves será privada y sólo la podrá usar su propietario y que la otra será pública y podrá ser vista por todos, incluso personas hostiles. El gran consumo de recursos de procesador de la criptografía asimétrica hace que su uso no sea apto para la encriptación de grandes cantidades de datos, pero la amplia cantidad de posibilidades que ofrecen las claves públicas y privadas la convierten en una herramienta imprescindible. La clave pública servirá para encriptar pequeñas cantidades de datos (como otras claves, simétricas por ejemplo), mientras que la clave privada se utilizará para operaciones de firma digital (se firma con algo que sólo pertenezca a uno mismo).

b. Certificados digitales X509

Aunque los algoritmos utilizados comúnmente en Internet y en las empresas se consideran fiables, el análisis de sistemas criptográficos muestra que la vulnerabilidad está sobre todo en los riesgos asociados a la transmisión de la clave pública de un usuario o un servidor. El mismo diseño de la criptografía asimétrica hace que esta clave sea absolutamente inútil por sí sola y que no haya modo alguno de deducir la clave privada a partir de la pública. Sin embargo, los fallos de seguridad en comunicaciones confidenciales o en las firmas digitales de documentos se basan en claves públicas cuyo nombre del propietario ha sido usurpado. Es decir, una clave pública circula, pero no es la verdadera y un farsante hace pasar su clave pública por la de otro. La persona engañada podría entonces encriptar datos muy confidenciales con la clave falsa y, por lo tanto, poner esta información en peligro.

Los certificados digitales X509 tienen como objetivo establecer de modo formal un enlace entre una identidad (nombre, dirección IP, etc.) y una clave pública. Los certificados no pueden falsificarse, ya que están firmados por un tercero con quien todos tienen una relación de confianza. Este tercer actor se llama "Certificate Authority" (autoridad de certificación). Pueden ser públicos y reconocidos por todo el mundo o privados y usados en un entorno restringido. En este caso, las aplicaciones cliente deberán configurarse para reconocer la autoridad de certificación que emitirá los certificados.

Cuando un servidor web tiene que usar un certificado, tiene que disponer de un certificado que demuestre su identidad: una clave pública relacionada con su nombre de host. Cuando se produce una conexión https por parte de un navegador, el servidor envía su certificado y el navegador comprueba su validez, debido a que el nombre con el que se accede al servidor es el que se anuncia en el certificado. Si no es así, el navegador muestra una alerta de seguridad. Entonces queda en manos del usuario cancelar la conexión segura o aceptarla, pero siempre con la duda de no saber si el servidor al que se dirige es legítimo o se trata de un farsante.

c. Generación local de un certificado

El funcionamiento de HTTP con SSL requiere que un certificado que contenga una clave pública del servidor web se envíe al navegador cliente y que esta clave pública se envíe siempre en un certificado. Por lo tanto, Apache configurado para SSL debe disponer de un certificado que podrá enviar a sus clientes.

El certificado utilizado por Apache podrá ser facilitado por una autoridad de certificación pública o

privada proporcionada por una aplicación especializada. En este supuesto, la autoridad proporcionará un certificado que se exportará en forma de archivo, se copiará en el disco del servidor Apache y éste lo utilizará.

Para un uso puntual o para realizar pruebas también se puede generar de manera local un certificado listo para usar que Apache podrá utilizar. Hay herramientas especializadas en esta función, pero están generalmente ligadas a una distribución, o se pueden crear a partir de todas las piezas generadas con las utilidades de la librería openssl. Para operaciones limitadas a las pruebas de la configuración ssl de Apache, se elegirá la solución más sencilla, que es utilizar la misma clave tanto para generar el certificado como para firmarlo.

Generación del certificado autofirmado

```
openssl req -x509 -nodes -newkey rsa:tamaño -keyout archivo_clave -out
archivo_certificado
```

Comando openssl para generar certificados: opciones y parámetros	
req	Solicitud de certificado.
-x509	Se desea un certificado autofirmado y no una petición de firma.
-nodes	La clave del servidor no debe estar protegida con contraseña.
-newkey rsa:*tamaño*	Se crean nuevas claves asimétricas RSA cuyo tamaño se proporciona en número de bits.
-keyout *archivo_clave*	El archivo que contiene la clave privada del servidor.
-out*archivo_certificado*	El archivo que contiene el certificado del servidor.

El comando anterior genera la solicitud del certificado. Los campos estándar del certificado se solicitan de forma interactiva al usuario. La mayoría de estos campos son informativos, pero para un certificado que se usa en un servidor web, el campo Common Name (nombre común) tiene que ser obligatoriamente el nombre DNS del servidor que aparecerá en la URL de acceso. En caso contrario, el navegador del cliente mostrará una alerta de seguridad cuando compruebe el certificado del servidor.

Ejemplo de generación de certificado

Es imprescindible informar correctamente el campo Common Name (CN). Es el que se asociará formalmente a la clave pública en el certificado digital.

```
root@servidor# openssl req -x509 -nodes -newkey rsa:1024 -keyout
servidor.key -out certificado.pem
Generating a 1024 bit RSA private key
......++++++
.................++++++
writing new private key to 'servidor.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:ES
State or Province Name (full name) [Some-State]:España
Locality Name (eg, city) []:Burgos
Organization Name (eg, company) [Internet Trials S.A.]:
Organizational Unit Name (eg, section) []:R&D
Common Name (eg, YOUR name) []:192.168.1.1
Email Address []:
root@servidor#
```

2. Configuración de ssl

a. Carga del módulo SSL

El módulo necesario para el funcionamiento de SSL es mod_ssl.

Carga del módulo

```
LoadModule ssl_module /ruta/mod_ssl.so
```

Donde *ruta* es la ruta absoluta del archivo de módulo. La ruta por defecto depende de la forma en que haya sido compilado Apache y, por lo tanto, de la distribución.

b. Configuración de las claves del servidor

A continuación hay que especificar al servidor cuáles son las claves que se deben usar para que funcione en modo SSL. Se tiene que disponer de su clave privada y de su certificado, que contendrá su clave pública.

Configuración de las claves

```
SSLCertificateFile /ruta/archivo_certificado
SSLCertificateKeyFile /ruta/archivo_clave
```

Archivo de configuración: Declaración de las claves de servidor	
SSLCertificateFile	Define el archivo que contiene el certificado del servidor.
SSLCertificateKeyFile	Define el archivo que contiene la clave privada del servidor.

c. Administración del funcionamiento en modo SSL

Sólo hay que solicitar al servidor que escuche por el puerto https y reiniciar el motor SSL que, una vez realizada la autentificación, permitirá encriptar las comunicaciones entre cliente y servidor.

Apertura del puerto https

```
Listen 443
```

Activación del motor SSL

```
SSLEngine on
```

Por supuesto, estos dos parámetros se deben añadir al archivo de configuración Apache.

d. Autentificación de los clientes mediante certificado

El funcionamiento en modo SSL estándar que se acaba de mostrar es el que se encuentra generalmente en Internet y sirve para la mayor parte de situaciones. Sin embargo, se pueden utilizar los certificados no para transmitir una clave de sesión encriptada y, por consiguiente, asegurar la confidencialidad; sino para garantizar la identidad de los clientes que se conecten al servidor. En esta configuración, el navegador cliente deberá disponer de un certificado que se comprobará por el servidor web. Para ello, el servidor Apache tiene que disponer de un certificado de la autoridad que haya emitido los certificados de los clientes.

Gestión de la autentificación mediante certificado en el archivo de configuración Apache

```
SSLVerifyClient require
SSLCACertificateFile certificado_ca
```

Donde *certificado_ca* representa el archivo de certificado de la autoridad que habrá firmado los certificados clientes. Los certificados clientes se deben instalar en el repositorio de certificados del navegador web.

Servidor proxy

1. Servidores proxy

Un servidor proxy se encarga de realizar una petición en nombre de un cliente a otro servidor mediante un protocolo determinado. El término español para proxy es también "intermediario", el servidor hace de intermediario para realizar una acción en nombre del cliente. Se dice que un servidor proxy trabaja en rotura de flujo. La mayoría de los proxys trabajan con el protocolo HTTP. Si se habla de proxy sin concretar el protocolo aplicado, se trata de un proxy web (HTTP).

a. Protección de clientes

Los servidores proxy son el único camino válido para ir a Internet (en principio, cualquier cliente tiene que pasar por el proxy para obtener contenido proveniente de Internet). Están en la primera línea en caso de que se produzca un ataque desde el exterior. Un servidor proxy correctamente configurado proporcionará, por tanto, una protección natural a los navegadores web en la red.

b. Servidores de caché

Todas las peticiones pasan por los proxy, el proxy obtiene los datos del servidor y los retransmite al cliente. En la mayoría de los casos, el servidor conserva en su disco una copia de estos datos para responder directamente a los sucesivos clientes que hagan las mismas peticiones. Por tanto, la navegación de los clientes se acelera porque no es necesario desviar continuamente las solicitudes a los servidores web.

La generalización de la banda ancha ha hecho que los beneficios de los servidores proxy de caché sean menos espectaculares.

c. Filtrado

Los servidores proxy ofrecen la posibilidad de rechazar parte o la totalidad de sus peticiones a algunos clientes. Entonces, se puede rechazar en bloque cualquier navegación o filtrar algunas url para impedir que se navegue en sitios no profesionales, por ejemplo. El servidor proxy es mejor que los cortafuegos para este tipo de tareas, ya que el proxy "entiende" qué es lo que se está solicitando (una URL), mientras que el cortafuegos se limita a autorizar o prohibir cualquier tráfico en un puerto (80 en el caso de http) sin diferenciar las páginas web visitadas.

d. Inconvenientes

Los servidores proxy no están libres de inconvenientes. Requieren una configuración específica de los clientes (hay que informar la dirección IP del proxy en el navegador) y se limitan a un protocolo de aplicación. Por tanto, se necesitan otros mecanismos de protección o de optimización para cada uno de los protocolos. Por ello, para cada despliegue planeado de un proxy, habrá que sopesar de forma precisa sus ventajas e inconvenientes.

2. El servidor proxy squid

a. Configuración básica

Squid se compone de un servicio cuyo script de inicio estándar va a buscar su configuración en un único archivo llamado **squid.conf**, que generalmente se ubica en **/etc/squid**.

La configuración de squid en un modo de funcionamiento estándar (ruptura de flujo de los accesos cliente y de algunas listas de acceso para gestionar los permisos) no es nada difícil, pero en la mayoría de las implementaciones el archivo de configuración por defecto de squid es impresionante. En el paquete proporcionado con debian, por ejemplo, el archivo ocupa un total de 5000 líneas, del que aproximadamente el uno por ciento se lee en el arranque del servicio.

Archivo squid.conf de un paquete debian sin comentarios

Se puede comprobar que hay un número considerable de listas de control de acceso (acl) así como de reglas definidas por defecto. Aún así, no evita que sea recomendable una configuración más precisa del proxy para mejorar la seguridad.

Sin una

```
# grep ^[^#] squid.conf
acl all src all
acl manager proto cache_object
acl localhost src 127.0.0.1/32
acl to_localhost dst 127.0.0.0/8
acl localnet src 10.0.0.0/8  # RFC1918 possible internal network
acl localnet src 172.16.0.0/12   # RFC1918 possible internal network
acl localnet src 192.168.0.0/16  # RFC1918 possible internal network
acl SSL_ports port 443    # https
acl SSL_ports port 563    # snews
acl SSL_ports port 873    # rsync
acl Safe_ports port 80    # http
acl Safe_ports port 21    # ftp
acl Safe_ports port 443         # https
acl Safe_ports port 70    # gopher
acl Safe_ports port 210         # wais
acl Safe_ports port 1025-65535  # unregistered ports
acl Safe_ports port 280         # http-mgmt
acl Safe_ports port 488         # gss-http
acl Safe_ports port 591         # filemaker
acl Safe_ports port 777         # multiling http
acl Safe_ports port 631         # cups
acl Safe_ports port 873         # rsync
acl Safe_ports port 901         # SWAT
acl purge method PURGE
acl CONNECT method CONNECT
http_access allow manager localhost
http_access deny manager
http_access allow purge localhost
http_access deny purge
http_access deny !Safe_ports
http_access deny CONNECT !SSL_ports
http_access allow localhost
http_access deny all
icp_access allow localnet
icp_access deny all
http_port 3128
hierarchy_stoplist cgi-bin ?
access_log /var/log/squid/access.log squid
refresh_pattern ^ftp:        1440    20%    10080
refresh_pattern ^gopher:     1440    0%     1440
refresh_pattern -i (/cgi-bin/|\?)  0 0%     0
refresh_pattern (Release|Package(.gz)*)$    0       20%    2880
refresh_pattern .            0       20%    4320
acl shoutcast rep_header X-HTTP09-First-Line ^ICY\s[0-9]
upgrade_http0.9 deny shoutcast
acl apache rep_header Server ^Apache
broken_vary_encoding allow apache
extension_methods REPORT MERGE MKACTIVITY CHECKOUT
visible_hostname beta
hosts_file /etc/hosts
coredump_dir /var/spool/squid
```

configuración particular, un servidor proxy squid ejerce de manera natural como servidor de caché de oficio y protege las redes locales mediante una ruptura del tráfico (las peticiones de los navegadores no van por Internet, sino que se paran en el proxy que consultará a los servidores en su lugar).

Antes de que pueda funcionar, el servidor todavía requiere una configuración mínima.

Configuración mínima de un proxy squid en el archivo squid.conf

```
http_port número_de_puerto
cache_dir ufs directorio tamaño dir_nivel_1, dir_nivel_2
```

```
visible_hostname nombre_servidor
```

Archivo squid.conf: configuración básica	
número_de_puerto	El número de puerto en el que el servidor está a la escucha y que debe configurarse en los navegadores. El valor por defecto es 3128, y 8080 es un valor histórico muy común.
directorio	El directorio en el que se almacenarán los datos que se pondrán en caché.
tamaño	Tamaño máximo en MB para los datos puestos en caché. Valor por defecto: 100 MB.
dir_nivel_1	Número máximo de subdirectorios de primer nivel del directorio de caché. Valor por defecto: 16.
dir_nivel_2	Número máximo de subdirectorios de segundo nivel del directorio de caché. Valor por defecto: 256.
nombre_servidor	Nombre de host del servidor proxy. Este nombre aparece especialmente en los registros de actividad.

b. Gestión del acceso a clientes

A continuación se trata de especificar quién puede o quién no puede acceder a Internet a través del servidor proxy.

La primera etapa consiste en definir los hosts o conjuntos de hosts (grupos, redes) al que se aplicará la autorización. Estos grupos se crean con el nombre de ACL (*Access Control List*).

Definición de listas de control de acceso en el archivo squid.conf

```
acl nombre_lista tipo_acl A.B.C.D/M
```

Archivo squid.conf: definición de acl		
nombre_lista		El nombre de la lista creada. Valor alfanumérico cualquiera.
tipo_acl	src	Definición de direcciones origen.
	dst	Definición de direcciones destino.
A.B.C.D/M		Dirección de red y máscara de subred (número de bits de la máscara).
		Dirección de host y máscara de subred (número de bits de la máscara).
		Intervalo de direcciones: A.B.C.D-E.F.G.H/M (número de bits a 1 de la máscara).

Ejemplo de definición de acls

Cabe destacar la definición de la acl "all" que incluye a todas las redes posibles.

```
acl all src all
acl red_local src 192.168.1.0/24
acl servidores_prohibidos dst 172.11.5.2-172.11.5.5/24
```

Sólo falta decirle a squid qué tiene que hacer con estas acls.

Autorización de acl en el archivo squid.conf

```
http_access autorización nombre_acl
```

Archivo squid.conf: autorización de acl	
autorización	Autorización o denegación de la acl. Los dos valores posibles son allow y deny.

Ejemplos de

nombre_acl	El nombre de la lista que se autoriza o se deniega.

autorizaciones de acls

Cada acl se trata con un control de acceso allow o deny.

```
acl all src all
acl red_local src 192.168.1.0/24
acl servidores_prohibidos dst 172.11.5.2-172.11.5.5/24
http_access deny servidores_prohibidos
http_access allow red_local
http_access deny all
```

Se pueden definir acl en un archivo externo al archivo de configuración principal.

Integración de un archivo de acl en el archivo de configuración

```
acl nombre_acl "archivo_acl"
```

Donde *archivo_acl* representa la ruta absoluta del archivo que contiene las acls. Este archivo tiene que estar informado obligatoriamente entre comillas dobles.

Comprobación de los conocimientos adquiridos: preguntas/respuestas

Ponga a prueba sus conocimientos respondiendo a las preguntas siguientes. Estas preguntas no siempre esperan una respuesta cerrada. Las preguntas planteadas en la certificación, a pesar de que abordan los mismos temas, en su mayoría serán planteadas en forma de preguntas de opción múltiple o que requieran una respuesta corta, en palabras escritas en el teclado.

1. Preguntas

1. ¿Cuál es el interés del funcionamiento modular de Apache?
2. ¿Cómo se puede validar la sintaxis de un archivo de configuración Apache sin tener que iniciar el servicio?
3. ¿Qué fuente de información es casi imprescindible para el uso correcto de módulos Apache y de directivas de configuración de estos módulos?
4. En la mayoría de las situaciones comunes de un servidor con hosts virtuales, ¿cómo sabe el servidor a qué host virtual se está consultando de entre todos los disponibles?
5. En el contexto de una autentificación sencilla gestionada de manera local por un servidor Apache, ¿qué comando permite crear cuentas de usuario en la base de datos local?
6. ¿Qué se puede poner en un archivo .htaccess?
7. ¿En qué aspecto los archivos .htaccess dependen de la directiva AllowOverride?
8. En general, ¿para qué sirve un certificado digital X509?
9. ¿Por qué se dice que un servidor proxy tradicional funciona rompiendo el tráfico?
10. En un servidor proxy squid, ¿se pueden poner acls squid en archivos de definición anexos?

2. Respuestas

1. ¿Cuál es el interés del funcionamiento modular de Apache?

   Más allá de la simple visualización de páginas web, el servidor web Apache soporta decenas de funciones, algunas comunes y otras específicas. La naturaleza modular de Apache permite cargar solamente los módulos necesarios para su funcionamiento y, por tanto, tener un código ejecutable cargado más ligero.

2. ¿Cómo se puede validar la sintaxis de un archivo de configuración Apache sin tener que iniciar el servicio?

   Ejecutando el comando apache con la opción -t.

3. ¿Qué fuente de información es casi imprescindible para el uso correcto de módulos Apache y de directivas de configuración de estos módulos?

   Es difícil configurar correctamente un servidor Apache complejo sin la ayuda del sitio web de documentación oficial de Apache. En primer lugar porque hay una gran cantidad de directivas: más de 400 en la versión 2.2 y más de un centenar de módulos; y también porque Apache es un software vivo y sus módulos están evolucionando continuamente.

4. En la mayoría de las situaciones comunes de un servidor con hosts virtuales, ¿cómo sabe el servidor a qué host virtual se está consultando de entre todos los disponibles?

   El servidor consulta la url solicitada por el cliente y averigua qué nombre se ha usado para la solicitud. Es el método más común, utilizado normalmente por los servicios de alojamiento web y por los proveedores de servicio de Internet que pueden albergar en un solo servidor físico varias decenas de sitios web bajo la forma de hosts virtuales.

5. En el contexto de una autentificación sencilla gestionada de manera local por un servidor Apache, ¿qué comando permite crear cuentas de usuario en la base de datos local?

   El comando es htpasswd, que permite crear, modificar y eliminar cuentas de usuario. Observe que este comando es útil cuando se trata de una autentificación local sencilla, que se usa en los casos más simples, pero que este método representa solamente una de las posibles formas de autentificación disponibles en Apache.

6. ¿Qué se puede poner en un archivo .htaccess?

Las mismas directivas que se habrían podido poner en un contexto Directory. La configuración se encuentra entonces en un archivo localizado más cerca de los datos que en el archivo de configuración de Apache.

7 ¿En qué aspecto los archivos .htaccess dependen de la directiva AllowOverride?

No hay, en principio, ningún motivo para que el servidor Apache busque en cada uno de sus directorios un posible archivo de configuración adicional. Por consiguiente, la directiva AllowOverride es necesaria para indicar que se acepta la existencia de una configuración complementaria y que conviene buscar este archivo de forma automática.

8 En general, ¿para qué sirve un certificado digital X509?

Se le puede dar muchos usos, pero en esencia un certificado digital asocia una identidad a una clave pública. Esta identidad puede ser una referencia a un usuario, un nombre, una dirección IP, etc. Esta asociación está garantizada, ya que está firmada por una autoridad de certificación a la que todo el mundo le ha otorgado su confianza. Si tuviéramos que encontrar una analogía en la vida cotidiana, se podría hablar de una identidad que asocia un nombre a una foto (clave pública), estando firmada por una autoridad (la jefatura de policía).

9 ¿Por qué se dice que un servidor proxy tradicional funciona rompiendo el tráfico?

Trabajando con un servidor proxy (intermediario), el cliente se dirige al servidor proxy indicándole el servidor web objetivo. El servidor proxy captura la petición y, a su vez, envía una petición al servidor objetivo, pero en nombre del cliente. Por tanto, se tiene un flujo de datos del cliente al proxy y otro flujo de datos del proxy al servidor objetivo. El proxy aprovecha la operación para realizar operaciones de filtrado, estadísticas, poner información en caché, etc.

10 En un servidor proxy squid, ¿se pueden poner acls squid en archivos de definición anexos?

Sí, la directiva acl usada en el archivo de configuración squid puede hacer referencia directamente a un parámetro de red o designar un archivo que contenga este mismo parámetro.

Trabajos prácticos

Para cubrir las necesidades internas de la empresa, se le pide desplegar dos servidores web para la intranet de la empresa. Uno contendrá datos accesibles para todos, incluyendo los visitantes de la empresa, y el otro contendrá datos confidenciales y sólo deberán acceder a él una serie de usuarios. Además, por temor a escuchas indiscretas, se le pide que aplique alguna medida para que no se puedan capturar datos sensibles por la red. Por lo tanto, planea aplicar una medida de protección usando SSL.

A pesar de disponer de dos servidores físicos, ansioso por economizar esfuerzos, decide implementar los dos servidores web en uno solo físico. Elige para esta tarea el servidor beta.

1. Configuración de un servidor web con dos sitios virtuales

a. Gestión de nombres DNS

Sus sitios web se diferenciarán gracias a la URL utilizada para acceder. Por tanto, necesitamos dos nombres distintos que hagan referencia a la misma IP.

Comandos útiles

- rndc
- vi

Operaciones

1. Crear en el dominio DNS pas.net del servidor alfa los dos registros, publico y privado, de tipo CNAME, que tengan como destino beta.pas.net.
2. Incrementar el número de versión del archivo.
3. Recargar la zona.
4. Alternativemente: si su servidor DNS no es operacional, también puede crear dos entradas en el archivo hosts de la estación de trabajo.

Resumen de los comandos y resultado por pantalla

Archivo /etc/hosts en el cliente:

Archivo

```
127.0.0.1 localhost
192.168.200.102 publico.pas.net privado.pas.net
```

/etc/bind/db.pas.net modificado en alfa:

```
$TTL    86400
pas.net.   IN  SOA      alfa.pas.net. root.pas.net. (
        8
        604800
        86400
        2419200
        86400 )

pas.net.    IN     NS      alfa.pas.net
alfa.pas.net.   IN   A          192.168.200.101
beta.pas.net.   IN   A          192.168.200.102
servidor-a      IN   CNAME      alfa.pas.net.
alpha           IN   CNAME      alfa
publico         IN   CNAME      beta
privado         IN   CNAME      beta
```

Recarga de datos de zona:

```
alfa:/etc/bind# rndc reload
server reload successful
alfa:/etc/bind#
```

b. Gestión de contenidos

Usted no se encarga del contenido de los sitios web. Por tanto, creará dos elementos simbólicos que le permitirán comprobar que el acceso a los distintos sitios está correctamente diferenciado.

Comandos útiles

- mkdir
- vi

Operaciones

1. Crear el directorio **/var/web/publico** para el sitio web publico.
2. Crear el directorio **/var/web/privado** para el sitio web privado.
3. Crear en cada uno de estos directorios un archivo **index.html** como el del siguiente ejemplo. Estos archivos deberán poder identificarse para saber si se está en el sitio privado o público.

```
<html>
  <body>
    <h1>Contenido del sitio</h1>
  </body>
</html>
```

Resumen de los comandos y resultado por pantalla

c.

```
[root@beta ~]# mkdir -p /var/web/publico
[root@beta ~]# mkdir -p /var/web/privado
[root@beta ~]#
[root@beta ~]# echo "<html><body><h1>Contenido público - acceso libre</h1></body></html>" >
/var/web/public/index.html
[root@beta ~]# echo "<html><body><h1>Contenido privado - acceso controlado</h1></body>
</html>" > /var/web/privado/index.html
[root@beta ~]#
```

Generación de un archivo de configuración simple

La complejidad del archivo de configuración proporcionado con el paquete le impresiona un poco. Descartando hacer cosas sin comprenderlas, decide dejar este archivo a un lado por el momento y hacer todas las pruebas de configuración con un archivo compuesto por varias piezas. El uso del archivo estándar se podrá hacer una vez se hayan realizado todas las pruebas.

Su objetivo por el momento es crear un servidor web que responda a peticiones HTTP.

Comandos útiles

- httpd
- useradd
- vi

Directivas de apache útiles

- ServerRoot
- User
- Group
- ErrorLog
- Listen
- DocumentRoot
- LoadModule
- DirectoryIndex
- ServerName

Operaciones

1. Crear la cuenta de usuario **apache-user**.
2. Crear el archivo **/etc/httpd/httpd.conf**.
3. Indicar al servidor que la base de datos de configuración se encuentra en el directorio **/etc/httpd**.
4. Indicar al servidor que los procesos deben detenerse por la cuenta de usuario**apache-user**.
5. Indicar al servidor que los procesos deben detenerse por el grupo **apache-user**.
6. Indicar al servidor que los errores deben guardarse en el archivo**/var/log/httpd/error.log**.
7. Indicar al servidor que la escucha de peticiones entrantes debe realizarse a través del puerto **80**.
8. Indicar al servidor que el contenido web se encuentra en el directorio**/var/web/public/**.
9. Indicar al servidor que su nombre principal es **192.168.200.102**.
10. Indicar al servidor que hay que cargar el módulo **/usr/lib/httpd/mod_dir.so** con el nombre **dir_module**.
11. Indicar al servidor que los archivos **index.html** tienen que mostrarse por defecto incluso si no se escriben en la URL.
12. Validar la sintaxis del archivo de configuración detallando adecuadamente que es este archivo de configuración el que desea comprobar y no el archivo proporcionado con el paquete.
13. Iniciar el servidor web sin usar el script de gestión del servicio y precisando que se quiere usar este archivo de configuración personal (/etc/httpd/httpd.conf).
14. Comprobar el acceso desde la estación de trabajo. Dependiendo del navegador que se utilice, puede que la página web rudimentaria no se muestre adecuadamente. Sólo hay que tener en cuenta el contenido.

Resumen de los comandos y resultado por pantalla

Creación de la cuenta de servicio:

Archivo

```
[root@beta conf]# useradd apache-user
[root@beta conf]#
```

/etc/httpd/httpd.conf

```
ServerRoot /etc/httpd
User apache-user
Group apache-user
Errorlog /var/log/httpd/error.log.
Listen 80
DocumentRoot /var/web/publico
ServerName 192.168.200.102
LoadModule dir_module modules/mod_dir.so
DirectoryIndex index.html
```

Comprobación de la sintaxis e inicio del servidor:

d.

```
[root@beta httpd]# httpd -f /etc/httpd/httpd.conf -t
Syntax OK
[root@beta httpd]# httpd -f /etc/httpd/httpd.conf -k start
[root@beta httpd]# pgrep -l http
3530 httpd
3531 httpd
3532 httpd
3533 httpd
3534 httpd
3535 httpd
[root@beta httpd]#
```

Adaptación para gestionar sitios virtuales

Animado por este éxito, decide implementar la gestión de sitios virtuales para que el servidor devuelva contenidos distintos según el nombre mediante el cual se acceda al mismo.

Comandos útiles

- httpd
- vi

Directivas apache útiles

- NameVirtualHost
- VirtualHost

Operaciones

1. Detener el daemon httpd antes de modificar el archivo de configuración.
2. Indicar al servidor que se encargará de gestionar hosts virtuales en todas las interfaces posibles a través del puerto 80.
3. Crear dos estructuras de sitios virtuales que responderán en todas las interfaces posibles a través del puerto 80.
4. Informar para cada uno de los sitios virtuales el nombre del servidor asociado (**publico.pas.net** y **privado.pas.net**).
5. Informar para cada uno de los sitios virtuales el directorio del contenido web asociado (**/var/web/publico** y **/var/web/privado**).
6. Comprobar la sintaxis del archivo de configuración.
7. Iniciar el daemon httpd con este archivo de configuración.
8. Desde el cliente, comprobar el acceso desde un navegador web a la url http://publico.pas.net.
9. Desde el cliente, comprobar el acceso desde un navegador web a la url

http://privado.pas.net.

Resumen de los comandos y resultado por pantalla

Parada del daemon httpd:

```
[root@beta httpd]# httpd -f /etc/httpd/httpd.conf -k stop
[root@beta httpd]# pgrep -l http
[root@beta httpd]#
```

Archivo

/etc/httpd/httpd.conf modificado:

```
ServerRoot /etc/httpd
ServerName 192.168.200.102
User apache-user
Group apache-user
Errorlog /var/log/httpd/error.log.
Listen 80
DocumentRoot /var/web/publico
LoadModule dir_module modules/mod_dir.so
directoryIndex index.html
NameVirtualHost *:80
<VirtualHost *:80>
 ServerName publico.pas.net
 DocumentRoot /var/web/publico
</VirtualHost>
<VirtualHost *:80>
 ServerName privado.pas.net
 DocumentRoot /var/web/privado
</VirtualHost>
```

2.

Control de acceso mediante contraseña en un sitio con ssl

a. Generación de certificados

Ahora es el turno de proteger el acceso al sitio privado contra las escuchas espía. Va a crear los certificados digitales necesarios para el funcionamiento con SSL. Siempre que no haya acceso público a este sitio web, se pueden utilizar certificados generados de manera local (y de forma gratuita).

Comandos útiles

- openssl

Operaciones

1. En el directorio /etc/httpd, generar los dos archivos **beta.key** y **certificado.pem**que se corresponden respectivamente con la clave privada del servidor beta y con su clave pública presentada en forma de certificado autofirmado. Las claves generadas deberán ser de 1024 bits y el nombre asociado al certificado tendrá que ser **privado.pas.net**.

Resumen de los comandos y resultado por pantalla

b.

```
[root@beta httpd]# openssl req -x509 -nodes -newkey rsa:1024
-keyout beta.key -out certificado.pem
Generating a 1024 bit RSA private key
...................++++++
..++++++
writing new private key to 'beta.key'
```

```
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [GB]:ES
State or Province Name (full name) [Berkshire]:Castilla y León
Locality Name (eg, city) [Newbury]:Burgos
Organization Name (eg, company) [My Company Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname) []:privado.pas.net
Email Address []:
[root@beta httpd]#
```

Configuración de SSL

Comandos útiles

- vi

Directivas útiles

- Listen
- LoadModule
- NameVirtualHost
- SSLCertificateFile
- SSLCertificateKeyFile
- SSLEngine

Operaciones

1. Indicar al servidor que debe cargar el módulo **mod_ssl.so** con el nombre**ssl_module**.
2. Indicar al servidor que debe utilizar el archivo **certificado.pem** como certificado que se entregará a los navegadores.
3. Indicar al servidor que debe utilizar el archivo **beta.key** como archivo de clave privada.
4. Indicar al servidor que va gestionar un host virtual también para el puerto **443**.
5. Indicar al servidor que el sitio virtual privado está ahora accesible a través del puerto **443**.
6. Indicar al servidor que debe estar a la escucha por el puerto **443** y activar el funcionamiento de SSL para el sitio virtual privado.
7. Reiniciar el servicio de Apache.
8. Comprobar el acceso en SSL desde la estación de trabajo Ubuntu. No sorprenderse con el aviso de seguridad. Decir que se entienden los riesgos, añadir una excepción, obtener el certificado y finalmente confirmar la excepción de seguridad.

Resumen de los comandos y resultado por pantalla

Archivo httpd.conf modificado:

c. Gestión de la autentificación

```
(...)
NameVirtualHost *:80

<VirtualHost *:80>
 ServerName publico.pas.net
 DocumentRoot /var/web/publico
</VirtualHost>
NameVirtualHost *:443
<VirtualHost *:443>
 SSLEngine on
 ServerName privado.pas.net
 DocumentRoot /var/web/privado
</VirtualHost>
LoadModule ssl_module modules/mod_ssl.so
SSLCertificateFile certificado.pem
SSLCertificateKeyFile beta.key
Listen 443
```

Finalmente, el acceso al sitio privado está protegido mediante SSL y sólo falta proteger el acceso a la parte confidencial mediante una autentificación por contraseña. Para sus primeros intentos decide configurar el control de acceso para un solo directorio (/var/web/privado/auth) y ubicar las directivas de configuración en un archivo oculto .htaccess en este directorio.

Comandos útiles

- chmod
- chown
- htpasswd
- vi

Directivas útiles

- AuthName
- AuthType
- AuthUserFile
- LoadModule
- Require

Operaciones

1. Crear un archivo de contraseñas de Apache **/etc/httpd/passwd** con una cuenta de usuario válida.
2. Asignar este archivo a las únicas cuentas de usuario y grupos de servicio Apache.
3. Gestionar los permisos de acceso a este archivo para que ninguna otra cuenta de usuario tenga acceso.
4. Indicar al servidor que debe cargar el módulo **/usr/lib/httpd/mod_auth_basic.so**con el nombre **auth_basic_module**.
5. Indicar al servidor que debe cargar el módulo **/usr/lib/httpd/mod_authz_user.so**con el nombre **authz_user_module**.
6. Indicar al servidor que debe cargar el módulo **/usr/lib/httpd/mod_authn_file.so**con el nombre **authn_file_module**.
7. En el directorio que se desea proteger (/var/web/privado/auth), crear el archivo**.htaccess** que contenga las directivas necesarias para la autentificación mediante archivo de contraseñas.

8. Reiniciar el servicio Apache.

9. Comprobar el acceso al sitio protegido desde la estación de cliente Ubuntu. Hay que conectarse usando SSL y tiene que aparecer una solicitud de autentificación.

Resumen de los comandos y resultado por pantalla

Gestión del archivo de contraseñas:

Añadir al archivo

```
[root@beta httpd]# htpasswd -c /etc/httpd/passwd usuario
New password: ********
Re-type new password: ********
Adding password for user usuario
[root@beta httpd]#
[root@beta httpd]# chown apache-user:apache-user passwd
[root@beta httpd]# chmod 440 passwd
[root@beta httpd]#
[root@beta httpd]# ls -l passwd
-r--r----- 1 apache-user apache-user 19 ago  2 10:46 passwd
[root@beta httpd]# cat passwd
usuario:2.eT/SXrEPV3E
[root@beta httpd]#
```

httpd.conf:

Archivo

```
(...)
LoadModule ssl_module modules/mod_ssl.so
SSLCertificateFile certificado.pem
SSLCertificateKeyFile beta.key
Listen 443

LoadModule auth_basic_module modules/mod_auth_basic.so
LoadModule authz_user_module modules/mod_authz_user.so
LoadModule authn_file_module modules/mod_authn_file.so
```

.htaccess:

3.

```
authType basic
AuthName "Se necesita comprobar sus credenciales"
Require valid-user
AuthUserFile /etc/httpd/passwd
```

Despliegue de un servidor proxy en el servidor alfa

Después de haber gestionado los servidores web, es momento de ocuparse de los clientes. Para proteger y optimizar el funcionamiento de la navegación por Internet decide desplegar un servidor proxy en el servidor alfa. Este servidor cumplirá dos objetivos: pondrá en caché aquellos datos que se consulten frecuentemente y prohibirá el acceso a un sitio web de juegos en línea que causa furor actualmente en la empresa.

a. Instalación de los binarios

Instale el servidor squid en alfa con el comando siguiente:

```
apt-get install squid
```

b. Configuración básica

Para controlar perfectamente su implementación, decide crear una vez más un archivo de configuración desde cero.

Comandos útiles

- chown
- mkdir
- mv
- vi

Directivas útiles

- cache_dir
- http_port
- visible_hostname

Manipulaciones

1. Crear el directorio **/var/proxy** que albergará los datos puestos en caché.
2. Asignar este directorio a la cuenta de servicio squid.
3. En el directorio **/etc/squid**, crear una copia de seguridad del archivo **squid.conf**con el nombre **ini.squid.conf**.
4. Crear un nuevo archivo **/etc/squid/squid.conf**.
5. Indicar en el nuevo archivo de configuración que el proxy recibirá las consultas de los clientes por el puerto 8080.
6. Indicar que el directorio de caché será **/var/proxy** con un tamaño máximo de 500 MB, 16 directorios de primer nivel de caché y 256 directorios de segundo nivel de caché.
7. Indicar que el nombre del servidor proxy que aparecerá en los archivos de registro es **prox**.
8. No iniciar el servicio de momento.

Resumen de los comandos y resultado por pantalla

Creación del directorio de caché:

```
alfa:~# mkdir /var/proxy
alfa:~# chown proxy /var/proxy
alfa:~# ls -ld /var/proxy
drwxr-xr-x 2 proxy root 4096 ago  3 10:32 /var/proxy
alfa:~#
```

Copia de seguridad del archivo de configuración:

```
alfa:/etc/squid# pwd
/etc/squid
alfa:/etc/squid# ls
squid.conf
alfa:/etc/squid# mv squid.conf ini.squid.conf
alfa:/etc/squid#
```

Nuevo archivo squid.conf:

```
http_port 8080
```

c.

```
cache_dir ufs /var/proxy 500 16 256
visible_hostname prox
```

Declaración y autorización de las acls

Si a pesar de nuestras advertencias ha intentado iniciar el servicio proxy, obtendrá un mensaje de error indicando que la acl no está definida. En efecto, incluso si no se desea gestionar el filtrado de ningún modo, tiene que definirse como mínimo la acl all.

Por tanto, va a declarar la acl e indicar que todo el tráfico está autorizado para todo el mundo y una acl llamada fun para indicar la dirección IP del servidor de juegos en línea prohibido.

Directivas útiles

- acl
- http_access

Operaciones

1. Declarar la acl all correspondiente para todos los orígenes posibles.
2. Configurar el navegador de la estación de trabajo para que use el servidor alfa como servidor intermediario (proxy).
3. Iniciar el servicio squid en alfa.
4. Comprobar el acceso a un sitio cualquiera desde el cliente y comprobar que no se puede llegar a ningún sitio.
5. Indicar que todo el tráfico correspondiente a la acl all está autorizado.
6. Reiniciar el servicio squid para alfa.
7. Comprobar el acceso a un sitio cualquiera desde el cliente y comprobar que ahora funciona mucho mejor.
8. Declarar la acl fun correspondiente a la dirección IP de un sitio prohibido en destino.
9. Prohibir todo tipo de tráfico correspondiente a la acl fun.
10. Reiniciar el servicio squid en alfa.

Resumen de los comandos y resultado por pantalla

Archivo squid.conf modificado:

```
http_port 8080
cache_dir ufs /var/proxy 100 16 256
visible_hostname prox

acl fun dst 12.34.56.78
acl all src all
http_access deny fun
http_access allow all
```

Reinicio del servicio:

```
alfa:/etc/squid# /etc/init.d/squid stop
Stopping Squid HTTP proxy: squid.
alpha:/etc/squid# /etc/init.d/squid start
Starting Squid HTTP proxy: squidCreating squid cache structure (warning).
2011/08/03 20:11:18| Creating Swap Directories
.
```

d. Prueba

```
alfa:/etc/squid#
```

funcional

Configure el navegador de la estación de trabajo cliente para que use el proxy instalado en alfa con el puerto 8080.

1. Desde el navegador Firefox, desplegar el menú **Editar** y hacer clic en **Preferencias**.
2. Hacer clic en la pestaña **Avanzado** y en la subpestaña **Red** hacer clic en el botón**Configuración**.
3. Seleccionar la configuración manual del proxy y escribir como proxy HTTP la dirección IP del servidor alfa y el puerto 8080.
4. Navegar de manera normal.
5. Intentar realizar una conexión a la url http://12.34.56.78 y comprobar que el servidor proxy squid la rechaza.

Requisitos y objetivos

1. Requisitos

Los conocimientos adquiridos en la certificación LPI nivel 1, especialmente:

- ☒ Edición de archivos.
- ☒ Saber configurar un cliente de correo.

2. Objetivos

Al final de este capítulo, será capaz de:

- ☒ Conocer el funcionamiento del envío de correos por Internet.
- ☒ Conocer los principales MTA.
- ☒ Asegurar una configuración sencilla de Postfix.
- ☒ Configurar dominios virtuales con Postfix.
- ☒ Configurar los MDA courier-pop y courier-imap.
- ☒ Conocer el MDA Dovecot.

Los MTA

Los MTA (*Mail Transfer Agent*) son los servidores que se encargan del envío y la recepción de correo electrónico y constituyen la espina dorsal de cualquier sistema de correo en redes IP. Son los servidores que gestionan el correo electrónico en un determinado dominio de correo. Cada servidor de correo público es un MTA y todos los MTA se comunican entre ellos mediante el protocolo SMTP.

1. El protocolo SMTP

El protocolo SMTP (*Simple Mail Transfer Protocol*) se utiliza para transmitir correos electrónicos a servidores de correo. SMTP puede utilizarse desde un cliente de correo (Outlook, Thunderbird, etc.) para enviar un correo electrónico a su servidor de correo, pero también entre los servidores de correo del emisor y los del destinatario. Se ha visto en el capítulo Resolución de nombres DNS cómo los registros MX asociados a un nombre de dominio del destinatario permiten encontrar la dirección IP del servidor. Una vez llega a su destino, el mensaje se conserva hasta que el destinatario se dé cuenta de que tiene correo pendiente. La lectura del correo puede realizarse directamente en el servidor o a través de un MDA (*Mail Delivery Agent*) con un protocolo de recepción de correo (POP o IMAP).

SMTP usa una sintaxis básica fácilmente comprobable desde un cliente telnet o nc.

Ejemplo de uso por línea de comandos del protocolo SMTP

```
alfa:~# telnet 192.168.199.10 25
Trying 192.168.199.10...
Connected to 192.168.199.10.
Escape character is '^]'.
ehlo usuario.com
220 alfa.localdomain ESMTP Postfix
250-alfa.localdomain
250-PIPELINING
250-SIZE 1000
250-VRFY
250-ETRN
250-ENHANCEDSTATUSCODES
250-8BITMIME
250 DSN
MAIL FROM: <usuario@usuario.com>
250 2.1.0 Ok
RCPT TO: <usuario>
250 2.1.5 Ok
DATA
354 End data with <CR><LF>.<CR><LF>
.
Buenas
¿Qué tal estás?
.
250 2.0.0 Ok: queued as E264474E02
QUIT
221 2.0.0 Bye
Connection closed by foreign host.
alfa:~#
```

El comando ehlo se utiliza por defecto en todos los sistemas actuales y solicita al servidor mostrar las extensiones SMTP que soporta. Los sistemas más antiguos (anteriores a 2001) utilizan el comando helo.

2. Presentación de Sendmail

SendMail es el más antiguo y puede que el más famoso MTA utilizado en Internet de la historia. Se escribió

incluso antes de la creación del protocolo SMTP y, en ese periodo, los mensajes se transmitían usando FTP de un servidor a otro. No había lugar para la existencia de MDA ni de un protocolo de recepción de correo y cualquier correo recibido se tenía que leer directamente en el servidor.

Sendmail no es motivo de entusiasmo para todo el mundo por dos razones: como la fecha de su creación es de una época en que los riesgos de ataque se limitaban a chistes de escuela, se ha configurado a menudo sin importantes opciones de seguridad y, de este modo, ha participado en la retransmisión de millones de spam. Por otro lado, las dificultades derivadas de la configuración de Sendmail pueden desalentar a los más entusiastas.

Las múltiples evoluciones y reescrituras de Sendmail lo han convertido hoy en día en una herramienta en la que se puede depositar total confianza y es uno de los MTA más potentes y rápidos del mercado.

3. Presentación de Exim

Exim es un MTA relativamente reciente (sus primeros desarrollos datan de 1995) que tiene como objetivos la robustez y la flexibilidad. Es el MTA ofrecido por defecto en las distribuciones Debian y en la mayoría de sus derivadas.

4. Presentación de Postfix

Postfix es, en el mundo del open source, el MTA más popular y casi el más fácil de configurar. Muchos servicios de alojamiento y proveedores de servicios de internet utilizan Postfix para administrar las cuentas de correo de sus clientes.

El servidor SMTP Postfix

1. Configuración de Postfix

a. Gestión de cuentas

Un MTA tiene que gestionar las cuentas de correo de su dominio, lo que implica que el servidor tiene que gestionar la lista de usuarios que tienen una cuenta de correo en el dominio. Los MTA generalmente son capaces de usar bases de datos de usuarios de distinto tipo: archivos locales de cuentas locales, directorio ldap, bases de datos MySQL, etc.

La solución más simple, siempre disponible y que no necesita ninguna configuración particular, es utilizar directamente las cuentas de usuario del sistema Linux.

b. Gestión de alias

En general, la base de datos de cuentas de usuario utilizada por un MTA determina qué direcciones de correo son susceptibles de recibir correos electrónicos. Sin embargo, puede que un usuario tenga varias cuentas de correo. Por ejemplo, es frecuente que el administrador de una red deba responder a los correos dirigidos a postmaster@dominio.ext. Es incluso una recomendación de la RFC SMTP. Para este tipo de uso, un MTA utiliza una relación de correspondencias entre cuentas llamadas alias. Postfix utiliza el archivo de declaración de alias **/etc/aliases** y lo utiliza en una base de datos generada a partir del archivo de alias mediante el comando **postalias**.

Archivo de declaración de alias

```
# /etc/aliases
mailer-daemon: postmaster
postmaster: root
nobody: root
hostmaster: root
usenet: root
news: root
webmaster: root
www: root
ftp: root
abuse: root
noc: root
security: root
root: usuario
```

Cualquier modificación del archivo **/etc/aliases** debe ir acompañada de una redeclaración de la base de datos con el comando **postalias**.

Generación de la base de datos a partir del archivo

```
alfa:~# postalias /etc/aliases
alfa:~#
```

c. El comando postfix

Generalmente un script de configuración estándar inicia el servicio postfix. Sin embargo, se puede utilizar el comando postfix directamente, especialmente cuando se está en fase de pruebas y diagnóstico.

Utilización del comando postfix

```
postfix acción
```

Comando postfix: acciones comunes	
status	Muestra el estado funcional del servicio.
stop	Detiene el servicio de manera controlada. Los procesos en ejecución están autorizados para finalizar su ejecución.
start	Realiza comprobaciones e inicia el servicio.
check	Comprueba la validez del entorno de funcionamiento del servicio.
reload	Recarga la configuración. Preferible a un stop/start.
abort	Detiene el servicio inmediatamente. Los procesos en ejecución se detienen bruscamente.
flush	Intenta mandar todos los correos pendientes: los que ya han dado error y los que están a la espera de un nuevo intento.

d. Archivos de configuración

La configuración del servicio Postfix se encuentra en el archivo llamado **main.cf**, alojado generalmente en el directorio **/etc/postfix**.

```
myorigin = dominio_origen
mydestination = dominio_destino
mynetwork = red/máscara_de_bits
relayhost = relays_MTA
```

Archivo main.cf: parámetros principales	
dominio_origen	El que el servidor pone a continuación de la arroba (@) en el correo saliente. Puede ser distinto del dominio local inicialmente configurado. Es el dominio visto desde el exterior.
dominio_destino	El servidor gestiona correos con destino a este dominio. Puede ser idéntico al dominio de origen.
red/máscara_de_bits	El servidor permite reenviar los correos provenientes directamente de esta red. En principo la red local.
relays_MTA	Si se usa el parámetro relayhost, los correos se envían al exterior exclusivamente a través del MTA *relays_MTA*.

No es obligatorio usar el parámetro relayhost y, según el espíritu del protocolo SMTP, no debería ser necesario. Sin embargo, algunos proveedores de servicio de Internet bloquean cualquier tipo de tráfico SMTP saliente de sus redes si éste no se ha emitido por sus propios MTA. Por tanto, el parámetro relayhost hace posible que se dependa de un MTA externo para cualquier transmisión de correo.

Con un archivo de configuración mínimo y usando únicamente los parámetros mencionados anteriormente, un servidor postfix ya sería capaz de realizar sus tareas de MTA. Estando a la espera de que un cliente de correo los descargue a su destinatario (mediante un protocolo de obtención de correo electrónico como POP o IMAP), los mensajes se almacenan en el directorio **/var/mail** con el nombre del usuario destinatario.

Para comprobar el funcionamiento hasta el momento del proceso de configuración, se puede mandar un email desde un cliente SMTP (Outlook, Thunderbird, etc.) configurado para que utilice el servidor postfix como servidor SMTP. La lectura de mensajes a este nivel de configuración sólo puede hacerse desde una sesión shell en el servidor postfix con el comando **mail**. El comando **mail**se explica en la sección de clientes de correo. Los requerimientos de la certificación LPI establecen que hay que tener conocimientos básicos de este comando.

e. Comprobación de la configuración activa

Se puede comprobar la configuración real de un servidor postfix para detectar problemas importantes de funcionamiento (directorios inexistentes, etc.) y los parámetros aplicados por el servidor desde el archivo **main.cf**.

Comprobación del entorno funcional

```
postfix check
```

Parámetros reales

```
postconf -n
```

2. Gestión de dominios virtuales

En una configuración sencilla, un servidor postfix gestiona un solo dominio de correo: el que está asociado a la empresa o a la organización que lo alberga. Sin embargo, puede suceder que se desee gestionar varios dominios de correo. Esta tarea la cumplen a la perfección los dominios virtuales. Los servicios de alojamiento de Internet utilizan los dominios virtuales, con los que pueden gestionar varias centenas de dominios de clientes en un solo servidor. También se usan en empresas, donde el servicio informático gestiona el correo de dos entidades distintas cuando, por ejemplo, se realiza una adquisición.

a. Definición de dominios virtuales

Se ha visto anteriormente que el archivo **main.cf** tenía que albergar en la directiva **mydestination**el nombre del dominio de correo gestionado. Este dominio principal, coherente con el nombre completo del servidor, se llama dominio canónico. Si se desea administrar otros dominios habrá que declararlos primeramente usando la directiva **virtual_alias_domain**.

Declaración de dominios virtuales en main.cf

```
virtual_alias_domain dominio2, dominio3
```

Donde *dominio2* y *dominio3* representan los dominios virtuales gestionados por el servidor.

b. Gestión de usuarios para dominios virtuales

A continuación hay que especificar qué cuenta de usuario se asigna a qué cuenta de correo y para qué dominio. Esta asociación debe hacerse en un archivo cuyo nombre y ubicación estén especificados por la directiva **virtual_alias_maps** en el archivo de configuración **main.cf**. El nombre que se suele utilizar para este archivo es **/etc/postfix/virtual**.

Declaración del archivo de alias en main.cf

```
virtual_alias_maps = hash:/etc/postfix/virtual
```

Basta con crear a continuación el archivo de alias con el formato siguiente:

Formato del archivo de alias

```
dirección_de_correo1 cuenta_linux
dirección_de_correo2 cuenta_linux
```

Ejemplo de archivo de alias

```
root@servidor# cat /etc/postfix/virtual
toto@dominio.com toto
titi@dominio.com titi
usuario@dominio.com usuario
root@servidor#
```

Creación del archivo de alias en un formato que pueda utilizar postfix

```
postmap /etc/postfix/virtual
```

Este comando crea un archivo en formato Berkeley DB a partir del archivo de alias en texto plano.

Ejemplo de creación de un archivo de alias

El comando postmap crea el archivo virtual.db a partir del archivo en texto plano virtual.

```
alfa:/etc/postfix# cat virtual
usuario@otrodominio.com   usuario
alfa:/etc/postfix# postmap virtual
alfa:/etc/postfix# ls virtual*
virtual  virtual.db
alfa:/etc/postfix# file virtual.db
virtual.db: Berkeley DB (Hash, version 9, native byte-order)
alfa:/etc/postfix#
```

3. Gestión de cuotas

Se puede limitar el espacio de disco consumido por las cuentas de correo. Esta limitación se establece fácilmente mediante el parámetro **mailbox_size_limit** en el archivo de configuración. Del mismo modo, se puede limitar el tamaño de un mensaje con el parámetro **message_size_limit**.

Gestión de los tamaños máximos en main.cf

```
mailbox_size_limit = tamaño_máx_cuenta
message_size_limit = tamaño_máx_correo
```

Limitación del espacio de disco en main.cf	
tamaño_máx_cuenta	Límite de una cuenta de correo en bytes.
tamaño_máx_correo	Límite de un mensaje en bytes.

Recepción local de mensajes

Para un MTA, el otro objetivo principal es el de recibir correos con destino los usuarios de su dominio de correo. No hay nada pensado para la entrega del correo a los usuarios. La solución común es usar un MDA (*Mail Delivery Agent*) para que los mensajes puedan recibirse desde un MUA (*Mail User Agent*), llamado comúnmente cliente de correo. Mientras tanto, los archivos quedan almacenados localmente en el MTA.

1. El comando mail

En un modo de funcionamiento normal, un MTA tiene que gestionar el correo que llega del exterior y enviar el correo de salida, pero tareas como la redacción de correos o la lectura de los correos entrantes se realizan desde un cliente de correo con el que el usuario pueda trabajar con más comodidad. Sin embargo, hasta que se configure un cliente de correo para enviar correos y se instale un servidor de recepción de correo para recibir correo en los clientes, es práctico poder utilizar el comando tradicional **mail** directamente desde el servidor.

a. Envío de correos con el comando mail

El comando **mail** permite enviar correos electrónicos de forma bastante cómoda. Se puede redactar y enviar el correo con una línea de comando única, pero generalmente es más cómodo utilizar el comando de forma interactiva.

Fases para el envío de un correo mediante el comando mail

- Introduzca el comando mail seguido del nombre del destinatario. Puede ser el nombre simple de la cuenta de usuario o la dirección de correo del destinatario.
- En el intérprete de comandos, introduzca el asunto de su mensaje.
- A continuación, redacte su mensaje con tantas líneas como desee. No aparecerá ningún tipo de indicación para esta introducción de datos.
- Una vez haya terminado, en una nueva línea, introduzca el carácter punto "." únicamente en su línea.
- Si aparece la indicación "Cc:", introduzca si fuera necesario los destinatarios en copia. (Cc significa *Carbon copy* o copia de carbón como en la época en la que las fotocopiadoras no existían). Si no hay que incluir en copia a ningún destinatario, pulse simplemente enter.
- Su correo se ha enviado al MTA local y será procesado por él mismo.

Ejemplo de envío de correo con el comando mail

El uso del comando mail para su uso en pruebas permite ganar una cantidad de tiempo nada despreciable.

```
alfa:/home/usuario# mail toto
Subject: Comida hoy
Buenas,
Si quieres venir a comer hoy con nosotros dinos algo.

Usuario
.
Cc:
alfa:/home/usuario#
```

b. Lectura de correos con el comando mail

Más aún que para el envío de correos, el comando **mail** es útil para leer los correos recibidos sin necesidad de instalar un servicio de recepción de correo. En efecto, un cliente de correo puede enviar fácilmente un mail dirigiéndose directamente al MTA usando SMTP. En cambio, en lo que se refiere a la lectura de mensajes recibidos desde un cliente de correo, hay que poder dirigirse al servidor con un protocolo de recepción: POP o IMAP. Si no hay ningún servidor POP o IMAP instalado, el comando **mail** es la solución más práctica para leer correos.

Lectura de un correo recibido con el comando mail

- Introduzca el comando mail y compruebe la existencia de una lista de correos no leídos.
- Introduzca el número del mensaje que desea consultar.
- Después de leer el correo, salga de la interfaz pulsando q.

Ejemplo de uso del comando mail para consultar un mensaje recibido

```
toto@alfa:~$ mail
Mail version 8.1.2 01/15/2001.  Type ? for help.
"/var/mail/toto": 4 messages 4 new
>N  1 usuario@pas.net       Sun Aug  6 13:12   15/398   hola
 N  2 usuario@pas.net       Sun Aug  6 15:24   17/438   Comida hoy
 N  3 usuario@pas.net       Sun Aug  6 19:10   14/402   Hello
 N  4 usuario@pas.net       Sun Aug  6 19:10   14/412   Are you Chip or Dale ?
& 2
Message 2:
From usuario@pas.net  Sun Aug  6 15:24:11 2011
X-Original-To: toto
To: toto@pas.net
Subject: Comida hoy
Date: Sun,  6 Aug 2011 15:24:11 +0100 (CET)
From: usuario@pas.net (root)

Buenas,
Si quieres venir a comer hoy con nosotros dinos algo.

Usuario

& q
Saved 1 message in /home/toto/mbox
Held 3 messages in /var/mail/toto
toto@alfa:~$
```

2. Formatos mbox y maildir

Una vez que un mensaje ha sido recibido por un MTA, se tiene que almacenar a la espera de la recepción por parte de un usuario. Tradicionalmente existen dos formatos principales que permiten conservar estos mensajes de forma estructurada: mbox y Maildir.

a. Formato mbox

El formato mbox se utiliza para almacenar los mensajes recibidos por un usuario. Es un formato rudimentario y bastante antiguo, en el que todos los mensajes se concatenan y un único archivo contiene todos los correos recibidos. Este formato tiene como ventaja su simplicidad y se puede usar fácilmente, incluso con un simple editor de texto (basta con detectar el correo buscado en el contenido del archivo). El comienzo de un mensaje se detecta mediante la secuencia de caracteres**From** al principio de línea. En cambio, presenta limitaciones inherentes a su modo de funcionamiento. El acceso desde varios programas al archivo es muy peligroso, debido a que cualquier operación de escritura en un archivo en formato mbox desde dos programas distintos dejaría el archivo corrupto y, por consiguiente, conduciría a la pérdida de la cuenta de correo. Por ello, hay mecanismos de bloqueo para el archivo mbox, pero lamentablemente puede que haya programas distintos que no reconozcan el mismo mecanismo de bloqueo y terminaríamos de todos modos en una situación catastrófica. La solución a estos problemas la proporciona el formato maildir.

b. Formato maildir

El formato maildir utiliza una estructura de directorios para almacenar los correos recibidos para un usuario. A diferencia del formato mbox, maildir utiliza un archivo por correo recibido. Cualquier operación realizada en un mensaje no afecta, por tanto, al resto de datos.

Un directorio en formato maildir contiene tres subdirectorios: **tmp**, **new** y **cur**. Los correos se almacenan inicialmente en **tmp** y después se mueven a **new**. Finalmente, después de haber sido leídos por un programa

del usuario, los mensajes se mueven a **cur**. Los correos se guardan en su directorio de asignación con un nombre único sin relación alguna con el asunto del correo.

c. Utilización del formato maildir en postfix

Por defecto, postfix utiliza el formato mbox para almacenar los correos recibidos por los usuarios. Sin embargo, se puede hacer (y a menudo se recomienda) que utilice el formato maildir en su lugar. Esta operación se realiza simplemente mediante una declaración en el archivo **main.cf**. El directorio **Maildir** se creará entonces en el directorio personal del usuario con la recepción del primer correo.

Declaración del formato Maildir en el archivo main.cf

```
home_mailbox = Maildir/
```

El comando mail usa únicamente el formato mbox. Por lo tanto no se puede utilizar si se usa el formato maildir. En ese caso, los mensajes deberán obtenerse mediante el uso de algún medio compatible, como por ejemplo un servidor POP o IMAP que sea compatible con maildir.

3. Procmail

Se puede indicar al MTA que procese los mensajes entrantes antes de almacenarlos. Postfix puede designar a un programa de terceros para este propósito. El más conocido de todos ellos es procmail. Basta con solicitar a postfix que utilice procmail (es fácil) y configurarlo para que realice un procesado con los correos entrantes (un poco más difícil). Este procesado puede ser de reorganización (poner ciertos mensajes en directorios), de filtrado (rechazar mensajes que contienen palabras prohibidas) o incluso llamar a otro programa para aplicar un procesado más pesado que procmail no sabría hacer por sí solo.

a. Indicar a postfix que utilice procmail

Declaración de uso de procmail por parte de postfix en el archivo main.cf

```
mailbox_command = /usr/local/bin/procmail
```

b. Configurar procmail

La configuración completa de procmail sobrepasa el alcance de la certificación LPI nivel 2 y, por tanto, de este libro. Sin embargo, se pueden aplicar algunos ejemplos de configuración simples sin dificultad.

Procmail lee su configuración del archivo **.procmailrc** que se encuentra en el directorio local del usuario. Este archivo contiene reglas que se aplicarán secuencialmente para todo correo entrante. El procesado se detiene cuando se satisface alguna regla.

Formato de una regla en el archivo ~/.procmailrc

```
:0 flags
condición
acción
```

Archivo ~/.procmailrc: opciones y parámetros	
:0	Señal que marca el comienzo de una regla de procesado.
flags	Opcional. Sobre qué debe aplicarse la búsqueda. Valor H para la cabecera solamente, B para el cuerpo del mensaje.
condición	Expresión regular que permite aislar los correos que se verán afectados por la regla.
acción	Qué hacer con los mensajes seleccionados.

Ejemplos de reglas en el archivo ~/.procmailrc

En el ejemplo siguiente, la búsqueda se realiza sólo en la cabecera del mensaje (es el valor por defecto) y seleccionará los correos que contengan las palabras "From" al comienzo de la línea y la cadena de caracteres "toto" en la misma línea. La tercera línea de la condición moverá el correo al directorio amigos/toto en el directorio de correo (y por lo tanto al subdirectorio del buzón de entrada en el cliente de correo).

Para la

```
:0
* ^From.*toto
amigos/toto
```

impresión de todos los correos cuyo tamaño es inferior a 1000 bytes.

```
:0
* < 1000
| /usr/bin/lp
```

4. Alternativas al correo

Durante mucho tiempo, la cantidad de recursos que consumía la infraestructura de correo, tanto en espacio de disco como en ancho de banda en la red, era un problema para los administradores. Aparecieron comandos alternativos que permitían a los usuarios conectados comunicarse de forma independiente al sistema de correo y con un consumo de recursos mucho inferior.

a. write y wall

Se pueden enviar mensajes cortos con los comandos **write** y **wall**. El comando **write** permite enviar un mensaje a un usuario conectado, mientras que **wall** (write all) difunde el mensaje a todos los usuarios conectados.

Envío de mensajes con write

```
write nombre_usuario
(introducir el mensaje terminándolo con Ctrl-D)

write < archivo_mensaje
```

Donde *nombre_usuario* representa un usuario del sistema conectado a una sesión interactiva, y*archivo_mensaje* el archivo que contiene el texto que se enviará.

Difusión de un mensaje con wall

```
wall
(introducir el mensaje terminándolo con Ctrl-D)

wall < archivo_mensaje
```

b. issue e issue.net

El contenido del archivo **/etc/issue** se muestra antes de la solicitud de identificación local y ofrece una posible comunicación con los usuarios.

El contenido del archivo **/etc/issue.net** se muestra antes de la autentificación de un usuario que se conecte por telnet.

c. motd

El contenido del archivo **/etc/motd** (*Message Of The Day*) se visualiza después de la apertura de una sesión con éxito.

Recepción remota de mensajes

1. Funcionamiento conjunto de MTA, MDA y MUA

La función de un MTA (*Mail Transfer Agent*) en lo que a la recepción de mensajes respecta se limita a la obtención y almacenamiento de los mensajes entrantes. Para que un usuario pueda leer y trabajar cómodamente con su correo, tiene que usar un MUA (*Mail User Agent o cliente de correo*) que funciona con un protocolo de recepción de correo: POP o IMAP. Postfix no es más que un MTA y no gestiona estos protocolos. Por lo tanto hay que añadir un servicio MDA (*Mail Delivery Agent*) de recepción de correo para los usuarios. La certificación LPI establece que hay que conocer los servidores courier-pop, courier-imap y Dovecot.

Cuando un mensaje llega al MTA, desde un punto de vista MTA ya ha llegado a destino. Por tanto, el MTA lo guarda en su disco local, en nuestro caso en formato mbox o maildir. Si se instala un servidor POP o IMAP, su función será, después de haber identificado el usuario, buscar los mensajes entrantes en este espacio de almacenamiento y proporcionárselos al cliente de correo.

a. El protocolo POP3

El protocolo POP3 funciona a través del puerto 110 y usa TCP como protocolo de transporte. Descarga los mensajes desde un buzón de usuario a un cliente de correo. A continuación, normalmente se borran los correos del buzón, liberando espacio de disco del servidor. Sin embargo, cada vez es más frecuente configurar POP desde el cliente para que deje una copia de los correos en el servidor.

b. El protocolo IMAP4

El protocolo IMAP4 funciona a través del puerto 143 y usa TCP como protocolo de transporte. Descarga las cabeceras de los correos desde el servidor y el cliente decide a continuación la acción que desea realizar con estos mensajes: consultarlos, borrarlos, moverlos, etc. Los mensajes se conservan en el servidor, pero se puede configurar los clientes IMAP para que sincronicen los mensajes descargados para una consulta sin conexión.

2. Servidores Courier-IMAP y Courier-POP

Los servidores courier-pop y courier-imap pertenecen a una suite software llamada "Courier Mail Server". Esta aplicación se ha creado para proporcionar el conjunto de servicios comunes de gestión de correo electrónico, pero como es modular sus componentes suelen utilizarse solos para proporcionar un servicio concreto.

a. Formato de mensajes para los servicios courier

Los servicios courier-pop y courier-imap encontrarán los correos entrantes exclusivamente en un directorio en formato maildir. No son compatibles con el formato mbox. Por tanto, habrá que configurar postfix para que utilice el formato maildir.

b. Configuración de servicios

Ésta es la buena noticia: en principio sólo hay que instalar el servicio e iniciarlo. Los parámetros por defecto son adecuados para el modo de funcionamiento estándar. Los archivos de configuración generalmente se encuentran en el directorio **/etc/courier** y se llaman **pop3d** para el servicio POP e **imapd** para el servicio IMAP.

Si el directorio de almacenamiento de los correos en formato maildir no debe usar el nombre por defecto (Maildir), hay que especificar en estos archivos de configuración el nombre usado realmente.

Nombre del directorio maildir en el archivo de configuración pop3d o imapd

```
MAILDIRPATH=nombredirmaildir
```

Donde *nombredirmaildir* representa el directorio usado para almacenar los correos recibidos en formato maildir.

Si el servidor dispone de varias interfaces físicas, se puede limitar las interfaces de escucha del daemon imap.

Restricciones de la interfaz activa en el archivo de configuración pop3d o imapd

```
address = dirección_interfaz
```

Donde *dirección_interfaz* representa la dirección IP de la interfaz válida para establecer las conexiones con los clientes.

c. Validación de la autentificación

Cuando se utiliza Courier-POP o Courier-IMAP, un cliente de correo presenta el identificador y la contraseña del usuario del que se desea consultar el correo. La librería "courier", común a estos dos servicios, valida estos elementos de identificación. Puede ser útil comprobar por línea de comandos que la cuenta utilizada se autentifica correctamente con esta librería. La utilidad**authtest** sirve para esta tarea.

Comprobación de la validez de una cuenta con authtest

```
authtest usuario contraseña
```

Donde *usuario* y *contraseña* son las credenciales de autentificación que presenta el cliente de correo para conectarse usando imap o pop al servidor.

Ejemplo de uso de authtest

Hasta aquí va todo bien...

```
alfa:/etc/courier# authtest usuario password
Authentication succeeded.

     Authenticated: usuario  (system username: usuario)
    Home Directory: /home/usuario
           Maildir: (none)
             Quota: (none)
Encrypted Password: $1$YSIbmjnM$makfir51Gla3ZpfRq5dmu.
Cleartext Password: password
           Options: (none)
alfa:/etc/courier#
```

3. Servidor Dovecot

Dovecot es otro servidor de recuperación de correo que hay que conocer para la certificación LPI. Se ha desarrollado con el objetivo de proporcionar el máximo rendimiento y seguridad. Su despliegue es relativamente simple, pero es tan rico funcionalmente que las posibilidades de configuración son innumerables y, a menudo, desalentadoras.

Dovecot soporta de forma nativa los formatos de buzón mbox y maildir.

a. Configuración de Dovecot

El servidor Dovecot halla su configuración en el archivo **dovecot.conf**, generalmente ubicado en el directorio **/etc/dovecot**. Si el servicio tiene que usarse en una infraestructura sencilla y común, simplemente habrá que modificar su configuración para que acepte la autentificación por contraseña sin encriptar. Puede parecer sorprendente que no se protejan las contraseñas en un servidor de correo, pero en un uso tradicional cuando el mensaje circula por Internet tampoco se protege de modo alguno y es visible para todos. Asegurar entonces únicamente la etapa cliente/servidor daría una falsa sensación de seguridad del contenido del mensaje. La contraseña del cliente de correo ya no circularía sin encriptar, pero el mensaje sólo estaría protegido de los vecinos inmediatos. Sin embargo, se puede configurar el cliente de correo para utilizar los protocolos POP o IMAP sobre SSL, aportando confidencialidad al tramo cliente/servidor pero teniendo en mente que los mensajes han transitado hasta llegar al servidor sin ningún tipo de protección antes de llegar al servidor. La verdadera seguridad en el contenido de los mensajes sólo puede obtenerse con un protocolo que gestione de extremo a extremo, como SMIME.

Autorización de autentificaciones sin encriptar en el archivo dovecot.conf

```
disable_plaintext_auth = no
```

Esta línea puede añadirse en cualquier parte del archivo de configuración, pero generalmente está comentada en los archivos de configuración por defecto facilitados con los paquetes.

b. Visualización de la configuración

El número de parámetros definidos en el archivo **dovecot.conf** puede ser impresionante y hacer que su interpretación se vuelva difícil. Además, puede ser útil comprobar un parámetro de configuración sin tener que recorrer las decenas o centenas de líneas del archivo. El comando**dovecot** llamado con la opción **-a** permite ver los parámetros reales del servidor.

Ejemplo de uso del comando dovecot para visualizar la configuración

El resultado que se muestra a continuación está cortado.

```
alfa:/etc/dovecot# dovecot -a | wc -l
139
alfa:/etc/dovecot# dovecot -a | head -20
# 1.0.15: /etc/dovecot/dovecot.conf
base_dir: /var/run/dovecot
log_path:
info_log_path:
log_timestamp: %Y-%m-%d %H:%M:%S
syslog_facility: mail
protocols: imap imaps pop3 pop3s
listen: *
ssl_listen:
ssl_disable: no
ssl_ca_file:
ssl_cert_file: /etc/ssl/certs/dovecot.pem
ssl_key_file: /etc/ssl/private/dovecot.pem
ssl_key_password:
ssl_parameters_regenerate: 168
ssl_cipher_list:
ssl_verify_client_cert: no
disable_plaintext_auth: no
verbose_ssl: no
shutdown_clients: yes
alfa:/etc/dovecot#
```

Comprobación de los conocimientos adquiridos: preguntas/respuestas

Ponga a prueba sus conocimientos respondiendo a las preguntas siguientes. Estas preguntas no siempre esperan una respuesta cerrada. Las preguntas planteadas en la certificación, a pesar de que abordan los mismos temas, en su mayoría serán planteadas en forma de preguntas de opción múltiple o que requieran una respuesta corta, en palabras escritas en el teclado.

1. Preguntas

1. En el funcionamiento de un servidor de correo de tipo MTA, ¿a qué se llama generalmente un alias?
2. Postfix soporta el parámetro relayhost. ¿En qué circunstancias se requiere usarlo?
3. ¿Se puede gestionar con postfix un dominio de correo en una red local pero presentarlo al exterior con otro nombre más adecuado?
4. Cuando un servidor postfix recibe un mensaje, ¿cómo sabe que debe tratarlo personalmente y no reenviarlo?
5. ¿Se puede comprobar la validez de la configuración de un servidor postfix sin tener que iniciar el servicio?
6. En la sintaxis SMTP o cuando se envía un correo con el comando mail, ¿cómo se indica que se ha terminado la redacción del mensaje?
7. ¿Cómo se puede automatizar un procesado en los mensajes entrantes en un MTA?
8. Si un administrador quiere enviar un mensaje urgente a todos los usuarios conectados en modo consola (local, telnet o ssh), ¿dispone de alguna alternativa al envío de un mensaje con SMTP?
9. ¿Qué diferencia hay entre el contenido del archivo /etc/issue y el del archivo /etc/motd?
10. ¿Se puede comprobar la autentificación a través de los servidores courier (Courier-POP y Courier-IMAP) sin tener que configurar un cliente de correo?

2. Respuestas

1. En el funcionamiento de un servidor de correo de tipo MTA, ¿a qué se llama generalmente un alias?

   Es la asociación de una identidad con una cuenta existente. Por ejemplo, generalmente se debe poder enviar un mensaje a la dirección de servicio webmaster@sitio.com. Para que el webmaster no tenga que consultar su cuenta personal y la de webmaster, se crea un alias entre ambas identidades.

2. Postfix soporta el parámetro relayhost. ¿En qué circunstancias se requiere usarlo?

   La generalización del spam ha complicado un poco el envío de mensajes por Internet con el protocolo SMTP. Si la dirección IP pública asignada a una organización por su proveedor de servicios de Internet tiene un dudoso pasado, puede que la dirección en cuestión esté en una blacklist y, por tanto, los MTA correspondientes la rechazarán. Traspasar todo mensaje saliente a su proveedor de servicios de Internet para que se encargue de enviarlo es una solución interesante si no desea iniciar un procedimiento para eliminar la dirección IP de las blacklists.

3. ¿Se puede gestionar con postfix un dominio de correo en una red local pero presentarlo al exterior con otro nombre más adecuado?

   Sí, para ello hay que informar la directiva myorigin en el archivo de configuración de postfix. Es un comportamiento común, por ejemplo cuando una empresa cambia de nombre.

4. Cuando un servidor postfix recibe un mensaje, ¿cómo sabe que debe tratarlo personalmente y no reenviarlo?

   Compara el dominio de destino anunciado (los caracteres escritos a continuación de la arroba) con los definidos por la directiva mydestination en el archivo de configuración de postfix. Si son idénticos, el servidor postfix sabe que es el responsable de tratar el mensaje.

5. ¿Se puede comprobar la validez de la configuración de un servidor postfix sin tener que iniciar el servicio?

   Sí, con el comando postfix check. También se pueden visualizar los parámetros reales de la configuración con el comando postconf -n.

6 En la sintaxis SMTP o cuando se envía un correo con el comando mail, ¿cómo se indica que se ha terminado la redacción del mensaje?

Escribiendo una línea que consista únicamente en un punto. El punto tiene que ser el carácter único en la línea, validado inmediatamente con un retorno de carro.

7 ¿Cómo se puede automatizar un procesado en los mensajes entrantes en un MTA?

Llamando al comando de gestión de procmail. Se tiene que hacer referencia a procmail en el archivo de configuración postfix y poseer también su propio archivo de reglas.

8 Si un administrador quiere enviar un mensaje urgente a todos los usuarios conectados en modo consola (local, telnet o ssh), ¿dispone de alguna alternativa al envío de un mensaje con SMTP?

Sí, el antiguo comando wall se creó para ello. Envía un mensaje a todos los usuarios conectados. Se emplea frecuentemente antes de reiniciar el sistema o detener un servicio.

9 ¿Qué diferencia hay entre el contenido del archivo /etc/issue y el del archivo /etc/motd?

Los dos se muestran a los usuarios que se conecten al sistema, pero /etc/issue se muestra antes de la apertura de sesión -y por tanto todos los usuarios lo pueden ver- mientras que el contenido del archivo /etc/motd sólo es visible a través de una apertura de sesión con éxito.

10 ¿Se puede comprobar la autentificación a través de los servidores courier (Courier-POP y Courier-IMAP) sin tener que configurar un cliente de correo?

Sí, el comando authtest realiza esta comprobación. No determina un éxito funcional completo en producción, pero por lo menos comprueba que las librerías de autentificación de courier están instaladas y configuradas correctamente.

Trabajos prácticos

Con el fin de mejorar la comunicación dentro de la empresa, se le pide que despliegue un servicio de correo.

1. Gestión de los envíos

a. Instalación de un servidor postfix en el servidor alfa

Instale el servidor postfix en el servidor alfa escribiendo el comando siguiente:

```
apt-get install postfix
```

Si el asistente de instalación le realiza alguna pregunta, elija "Sin configuración" para indicar que desea realizar la configuración usted mismo.

Observe que la instalación de postfix incluye la eliminación del servicio de correo nativo Exim de las distribuciones Debian.

b. Configuración del servicio

Comandos útiles

- postconf
- postfix
- tail
- vi

Archivo útil

- main.cf

Operaciones

1. En el directorio **/etc/postfix**, crear el archivo **main.cf**.
2. En el archivo **main.cf**, indicar que los correos vendrán del dominio **pas.net**.
3. En el archivo **main.cf**, indicar que el servidor gestionará los correos con destino el dominio **pas.net**.
4. En el archivo **main.cf**, indicar la dirección de la red local.
5. Comprobar los parámetros reales de la configuración postfix.
6. Comprobar la coherencia de la configuración postfix.
7. Iniciar el servicio y comprobar que todo funcione correctamente.

Resumen de los comandos y resultado por pantalla

Archivo de configuración /etc/postfix/main.cf:

```
myorigin = pas.net
mydestination = pas.net
mynetwork = 192.168.200.0/24
```

Comprobación de los parámetros reales:

```
alfa:/etc/postfix# postconf -n
config_directory = /etc/postfix
mydestination = pas.net
```

```
myorigin = pas.net
alfa:/etc/postfix#
```

Comprobación de la coherencia de la configuración:

```
alfa:/etc/postfix# postfix check
alfa:/etc/postfix#
```

Inicio del servicio y comprobación:

```
alfa:/etc/postfix# /etc/init.d/postfix start
Starting Postfix Mail Transport Agent: postfix.
alfa:/etc/postfix# tail -1 /var/log/syslog
Aug 12 15:30:43 alfa postfix/master[5008]: daemon started --
version 2.5.5, configuration /etc/postfix
alfa:/etc/postfix#
```

c. Gestión de los alias postfix

Comandos y archivos útiles

- /etc/aliases
- postalias

Operaciones

1. Comprobar la presencia del archivo alias por defecto.
2. Crear la base de datos alias que deberá utilizar el servicio postfix a partir de su inicio.

Resumen de los comandos y resultado por pantalla

d.

```
alfa:~# cat /etc/aliases
# /etc/aliases
mailer-daemon: postmaster
postmaster: root
nobody: root
hostmaster: root
usenet: root
news: root
webmaster: root
www: root
ftp: root
abuse: root
noc: root
security: root
root: toto
alfa:~#
alfa:~# postalias /etc/aliases
alfa:~#
```

Integración DNS

Para que se puedan enviar mensajes desde otros MTA, decide crear un registro MX para hacer referencia a su dominio.

Comandos útiles

- rndc
- vi

Operaciones

1. Cree en el dominio DNS **pas.net** del servidor alfa un registro MX de prioridad 10 con MTA con el nombre **alfa.pas.net**.
2. Incremente el número de versión del archivo.
3. Recargue la zona.

Resumen de los comandos y resultado por pantalla

Archivo /etc/bind/db.pas.net modificado en alfa:

```
$TTL    86400
pas.net.   IN  SOA      alfa.pas.net. root.pas.net. (
        15
        604800
        86400
        2419200
        86400 )

pas.net.          IN      NS       alfa.pas.net.
pas.net.          IN      NS       beta.pas.net.
alfa.pas.net.     IN      A        192.168.200.101
beta.pas.net.     IN      A        192.168.200.102
servidor-a        IN      CNAME    alfa.pas.net.
client            IN      A        192.168.200.212
publico           IN      CNAME    beta
privado           IN      CNAME    beta
pas.net.          IN      MX 10    alfa.pas.net.
```

Recarga de datos de la zona:

```
alfa:/etc/bind# rndc reload
server reload successful
alfa:/etc/bind#
```

e. Envío y recepción de correos por línea de comandos desde el servidor alfa

Comandos útiles

- adduser
- mail
- su

Operaciones

1. En el servidor alfa, crear el usuario **titi** con la contraseña **password**.
2. En el servidor alfa, abrir un terminal con el usuario **usuario**.
3. Enviar un correo al usuario **titi**.
4. Abrir otro terminal con el usuario **titi**.
5. Comprobar sus mensajes.

Resumen de los comandos y resultado por pantalla

Añadir el usuario titi en alfa:

```
alfa:~# adduser titi
Añadiendo el usuario `titi' ...
Añadiendo el nuevo grupo `titi' (1002) ...
Añadiendo el nuevo usuario `titi' (1002) con grupo `titi' ...
Creando el directorio personal `/home/titi' ...
Copiando los archivos desde `/etc/skel' ...
Introduzca la nueva contraseña de UNIX:
Vuelva a escribir la nueva contraseña de UNIX:
passwd: contraseña actualizada correctamente
Cambiando la información de usuario para titi
Introduzca el nuevo valor, o presione ENTER para el predeterminado
  Nombre completo []: titi
  Número de habitación []:
  Teléfono del trabajo []:
  Teléfono de casa []:
  Otro []:
¿Es correcta la información? [S/n] S
alfa:~#
```

Envío de un correo por parte del usuario usuario:

```
usuario@alfa:~$ whoami
usuario
usuario@alfa:~$
usuario@alfa:~$ mail titi
Subject: Buenas
Solamente comprobar si funciona.

.
Cc:
usuario@alfa:~$
```

Comprobación de la recepción del correo por parte del usuario titi:

f. Paso de postfix al

```
usuario@alfa:~$ whoami
usuario
usuario@alfa:~$ su - titi
Contraseña:
titi@alfa:~$ whoami
titi
titi@alfa:~$ mail
Mail version 8.1.2 01/15/2001.  Type ? for help.
"/var/mail/titi": 1 message 1 new
>N  1 usuario@pas.net       Thu Aug 12 15:17   15/428   Buenas
& 1
Message 1:
From usuario@pas.net  Thu Aug 12 15:17:30 2010
X-Original-To: titi
To: titi@pas.net
Subject: Buenas
Date: Thu, 12 Aug 2010 15:17:30 +0200 (CEST)
From: usuario@pas.net (usuario)

Solamente comprobar si funciona.


& q
```

```
Saved 1 message in /home/titi/mbox
titi@alfa:~$
```

formato maildir

Archivo y comandos útiles

- /etc/postfix/main.cf
- vi

Operaciones

1. En el archivo de configuración, declarar el uso del formato maildir.
2. Reiniciar el servicio.

Resumen de los comandos y resultado por pantalla

Archivo main.cf modificado:

```
myorigin = pas.net
mydestination = pas.net
mynetwork = 192.168.200.0/24

home_mailbox = Maildir/
```

Reinicio del servicio:

```
alfa:/etc/postfix# /etc/init.d/postfix restart
Stopping Postfix Mail Transport Agent: postfix.
Starting Postfix Mail Transport Agent: postfix.
alfa:/etc/postfix#
alfa:/etc/postfix# tail -1 /var/log/syslog
Aug 12 15:49:43 alfa postfix/master[5101]: daemon started --
version 2.5.5, configuration /etc/postfix
alfa:/etc/postfix#
```

2.

Gestión de las recepciones

a. Instalación del servidor Courier-IMAP en el servidor alfa

Instalar el servidor Courier-IMAP en alfa escribiendo el comando siguiente:

```
alfa:~# apt-get install courier-imap
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes extras:
  courier-authdaemon courier-authlib courier-authlib-userdb courier-base
  expect fam libfam0 libltdl3 tcl8.4
(...)
```

Acepte todas las opciones por defecto en la ejecución del asistente de instalación.

b. Envío de un mensaje al usuario

El envío de este mensaje nos servirá para comprobar la correcta configuración del servidor IMAP.

Comandos útiles

- mail
- su

Operaciones

1. En el servidor alfa, abrir un terminal con el usuario **titi**.
2. Enviar un correo por línea de comandos al usuario **usuario**.

Resumen de los comandos y resultado por pantalla

Envío del correo por línea de comandos:

```
alfa:~# su - titi
titi@alfa:~$ whoami
titi
titi@alfa:~$ mail usuario
Subject: Hola usuario
Espero que el cliente de correo funcione correctamente.
.
Cc:
titi@alfa:~$
```

c. Gestión de correo desde la estación de trabajo

Para comprobar el funcionamiento del servidor imap, debe configurar un cliente de correo en la estación de trabajo.

El paquete Evolution es el cliente de correo por defecto pero puede utilizar cualquier tipo de cliente imap.

Comandos útiles

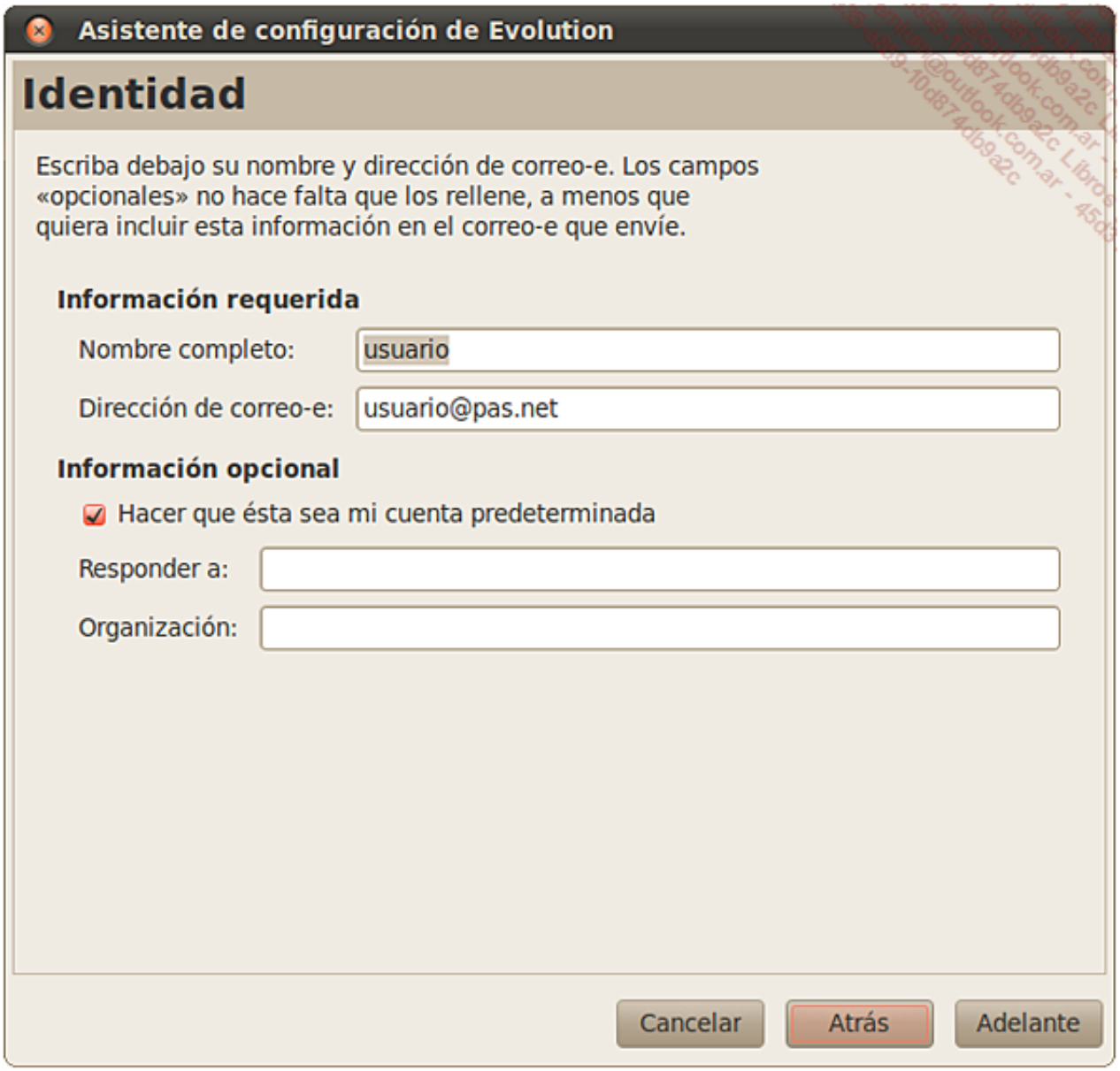
- Utilización de la interfaz gráfica.

Operaciones

1. En la estación de trabajo, iniciar el cliente **Correo y calendario de Evolution** desde el menú **Aplicaciones/Oficina**.
2. Utilizar todos los parámetros por defecto, a excepción de la identidad del usuario (usuario), el servidor IMAP (dirección IP o nombre DNS del servidor alfa) y el servidor SMTP (dirección IP o nombre DNS del servidor alfa).
3. Comprobar que aparece un mensaje en la ventana de Evolution.

Resumen de los comandos y resultado por pantalla

Configuración del cliente de correo Evolution:



Gestión de la identidad

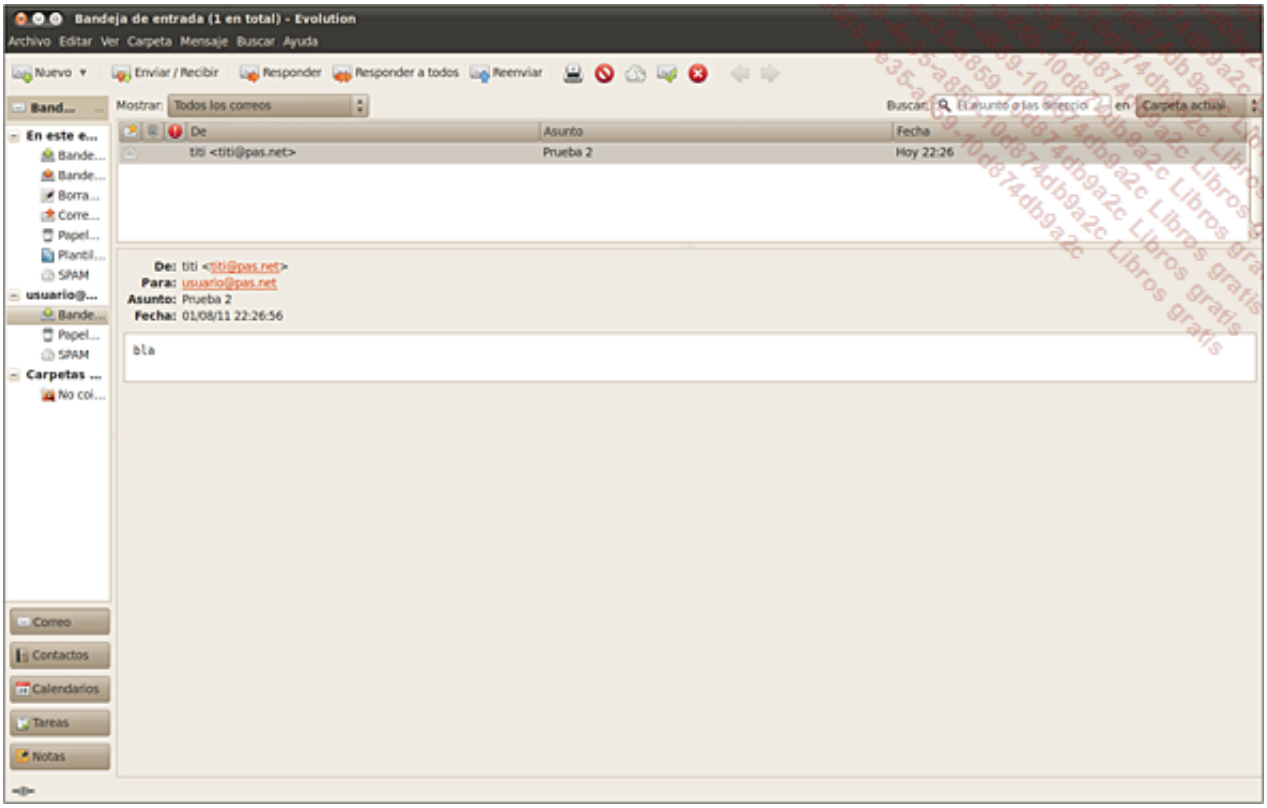
Configuración del servidor IMAP

Configuración del servidor SMTP

Envío de un correo desde la cuenta titi:

```
alfa:~# su - titi
titi@alfa:~$ mail usuario
Subject: Prueba 2
bla
.
Cc:
titi@alfa:~$
```

Visualización de mensajes en Evolution:



Observe que aunque el mensaje se mandó a usuario, aparece como si se hubiera mandado a usuario@pas.net. Este es el resultado de la correcta configuración de postfix (parámetro myorigin).

Requisitos y objetivos

1. Requisitos

Los conocimientos adquiridos con la certificación LPI de nivel 1, especialmente:

- ☒ Conocimientos básicos del direccionamiento IP.
- ☒ Conocimientos básicos del enrutamiento IP.
- ☒ Edición de archivos de texto.
- ☒ Conocimientos del archivo /etc/services.
- ☒ Conocimientos básicos del daemon inetd.

2. Objetivos

Al final del capítulo, será capaz de:

- ☒ Activar el enrutamiento en un servidor Linux.
- ☒ Añadir y quitar rutas estáticas.
- ☒ Configurar el filtrado con iptables.
- ☒ Configurar NAT para iptables.
- ☒ Configurar un cortafuegos Linux usando iptables.
- ☒ Mostrar la configuración de un cortafuegos existente.
- ☒ Modificar la configuración de un cortafuegos existente.
- ☒ Conocer los principales organismos de seguridad.
- ☒ Conocer las técnicas de análisis de los IP.
- ☒ Conocer el IDS Snort.
- ☒ Conocer el software de seguridad OpenVAS.

Enrutamiento y filtrado

1. Configuración de un servidor Linux como router

La función de enrutamiento se integra de forma nativa en el núcleo de Linux. Por lo tanto, hay pocas preguntas que hacerse, cualquier máquina Linux es un router en potencia. Sin embargo, esta función no está activada por defecto tras el arranque. Por tanto, hay que configurarla antes de realizar cualquier operación de enrutamiento.

a. Activación del enrutamiento en un servidor Linux

Sabemos que cualquier sistema Linux presenta un sistema de archivos virtual **/proc** que permite observar en directo un cierto número de componentes y parámetros. La activación del enrutamiento se realiza modificando el contenido del archivo **/proc/sys/net/ipv4/ip_forward**. Este archivo contiene un solo carácter que por defecto es **0** para indicar que el enrutamiento está inactivo.

Modificación del archivo ip_forward para activar el enrutamiento

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

Una vez se ha realizado el cambio, la máquina Linux está lista para enrutar paquetes que lleguen a sus interfaces. Este parámetro es volátil y se perderá una vez que la máquina se detenga. Sin embargo, se puede anular el enrutamiento realizando la operación inversa.

Modificación del archivo ip_forward para desactivar el enrutamiento

```
echo 0 > /proc/sys/net/ipv4/ip_forward
```

Otra posibilidad es usar el comando **sysctl** que permite modificar dinámicamente los parámetros funcionales del núcleo. **sysctl** permite modificar directamente todos los archivos que se encuentran en la estructura de directorios que cuelga de **/proc/sys**.

Activación del enrutamiento con sysctl

```
sysctl net.ipv4.ip_forward=1
```

Estos comandos son efectivos durante toda la duración de la sesión y deben volver a introducirse después de cada reinicio. Por supuesto, se pueden poner en un script de servicio llamado en el arranque, o modificar el archivo **/etc/sysctl.conf**.

Activación permanente del enrutamiento en el archivo /etc/sysctl.conf

```
net.ipv4.ip_forward = 1
```

b. Consulta de la tabla de enrutamiento

En este punto, el router Linux es perfectamente capaz de enrutar paquetes. Sin embargo, sólo podrá hacerlo a redes conocidas, es decir, referenciadas en su tabla de enrutamiento.

La tabla de enrutamiento se mantiene en memoria, pero se puede consultar usando varios comandos.

Visualización de la tabla de enrutamiento con el comando route

```
route -n
```

El parámetro -n es opcional, pero sirve para ahorrar mucho tiempo en la visualización, ya que no fuerza a que el comando intente resolver los nombres de las redes devueltas. Además, si la dirección en cuestión no se informa en una zona DNS inversa, esta petición se realiza en vano y hay que esperar varios segundos para que se muestre la salida del comando.

Visualización de la tabla de enrutamiento con el comando netstat

```
netstat -nr
```

Donde la opción -r hace que el comando muestre la tabla de enrutamiento y -n evita que se realice la resolución de nombres. El comando **netstat** tiene muchos posibles usos, pero a menudo se utiliza en este simple contexto de consulta de la tabla de enrutamiento.

Ejemplo de visualización de la tabla de enrutamiento

A menudo, la visualización de la tabla de enrutamiento es el único método sencillo para consultar el valor de la puerta de enlace predeterminada.

```
beta:~# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.1.0     0.0.0.0         255.255.255.0   U     0      0        0 eth1
192.168.0.0     0.0.0.0         255.255.255.0   U     0      0        0 eth0
0.0.0.0         192.168.0.1     0.0.0.0         UG    0      0        0 eth0
```

c. Gestión de rutas estáticas

Las únicas entradas presentes de forma automática en la tabla de enrutamiento son las redes a las que el router está conectado directamente, así como la puerta de enlace predeterminada. Por tanto, el router puede usar estas entradas de la tabla de enrutamiento sin necesidad de otra configuración. Si el router tiene que enrutar paquetes a otras redes, habrá que añadir de forma manual las rutas en la tabla de enrutamiento.

Agregar rutas estáticas en la tabla de enrutamiento

```
route add -net red_objetivo netmask máscara gw router
```

Agregar rutas estáticas: opciones y parámetros	
-net	La ruta incluida es una red. (El objetivo podría ser un solo host, aunque es menos frecuente.)
red_objetivo	La dirección de red que la nueva ruta permitirá alcanzar.
máscara	La máscara de subred asociada a la nueva ruta.
gw *router*	Indica el router que se utilizará para alcanzar la red objetivo.

Añadir una puerta de enlace por defecto

```
route add default gw router
```

```
route add -net 0.0.0.0 gw router
```

En la segunda sintaxis, 0.0.0.0 representa la ruta por defecto. Esta representación de la ruta por defecto es universal y se puede aplicar en casi la totalidad de sistemas que usen un tabla de enrutamiento IP.

Por supuesto, también se pueden eliminar las rutas estáticas que ya no sean necesarias o que estén guardadas por error.

Eliminación de rutas estáticas de la tabla de enrutamiento

```
route del -net red_objetivo netmask máscara
```

Ejemplo de adición de rutas

Ejemplo de eliminación de rutas

```
beta:~# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.1.0     0.0.0.0         255.255.255.0   U     0      0        0 eth1
192.168.0.0     0.0.0.0         255.255.255.0   U     0      0        0 eth0
0.0.0.0         192.168.0.1     0.0.0.0         UG    0      0        0 eth0
beta:~# route add -net 10.0.0.0 netmask 255.0.0.0 gw 192.168.1.99
beta:~# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.1.0     0.0.0.0         255.255.255.0   U     0      0        0 eth1
192.168.0.0     0.0.0.0         255.255.255.0   U     0      0        0 eth0
10.0.0.0        192.168.1.99    255.0.0.0       UG    0      0        0 eth1
0.0.0.0         192.168.0.1     0.0.0.0         UG    0      0        0 eth0
```

```
beta:~# route del -net 10.0.0.0 netmask 255.0.0.0
beta:~# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.1.0     0.0.0.0         255.255.255.0   U     0      0        0 eth1
192.168.0.0     0.0.0.0         255.255.255.0   U     0      0        0 eth0
0.0.0.0         192.168.0.1     0.0.0.0         UG    0      0        0 eth0
beta:~#
```

2. Iptables

Iptables se utiliza para gestionar el filtrado de paquetes IP en un sistema Linux. Usa un solo comando: **iptables**, y se configura mediante la aplicación de reglas de gestión de paquetes. Iptables puede filtrar el tráfico que transita por un router Linux, pero también el tráfico entrante y saliente de cualquier servidor o estación de trabajo en una sola interfaz.

Aunque iptables constituye una herramienta muy potente de gestión del tráfico, esta ventaja hace que su archivo de configuración sea todo excepto intuitivo. Sin embargo, con una aproximación estructurada se puede aprender su funcionamiento bastante rápido. Los siguientes párrafos exponen los conceptos fundamentales de iptables, para utilizarlos más tarde en configuraciones de cortafuegos.

a. Tablas

Iptables se basa en tablas asociadas a un modo funcional. Según el tipo de regla que se desea añadir en el funcionamiento de iptables, se precisará la tabla asociada. Las tablas principales utilizadas son **filter** para filtrar paquetes y **nat** para la traducción de direcciones entre una red privada y una red pública.

La tabla **filter** es la tabla por defecto. Por ello, cuando se establece una regla de iptables con el objetivo de filtrar paquetes, se sobreentiende y, por consiguiente, no se detalla.

La tabla **nat** sirve para traducir direcciones y debe precisarse sistemáticamente cuando se invoca.

b. Cadenas

Una cadena de iptables representa un tipo de tráfico desde el punto de vista de su circulación por una máquina. Las cadenas permiten especificar si una regla debe aplicarse al tráfico que entra en una máquina, que sale o que la cruza.

La cadena **INPUT** identifica el tráfico entrante, la cadena **OUTPUT** identifica el tráfico saliente y la cadena **FORWARD** identifica el tráfico que atraviesa la máquina, entrando por una interfaz y saliendo por otra. Atención, aunque un paquete que atraviesa el router es desde un punto de vista físico respectivamente entrante, encaminado y saliente, iptables lo considera solamente como encaminado (cadena FORWARD). Las cadenas INPUT y OUTPUT se reservan al tráfico con origen o destino explícito el host al que se le aplican las reglas.

También hay otra cadena llamada **POSTROUTING** que se utiliza en la configuración de NAT y tiene como objetivo tratar paquetes después de una operación de enrutamiento.

Las cadenas siempre se indican en mayúsculas en la sintaxis de iptables.

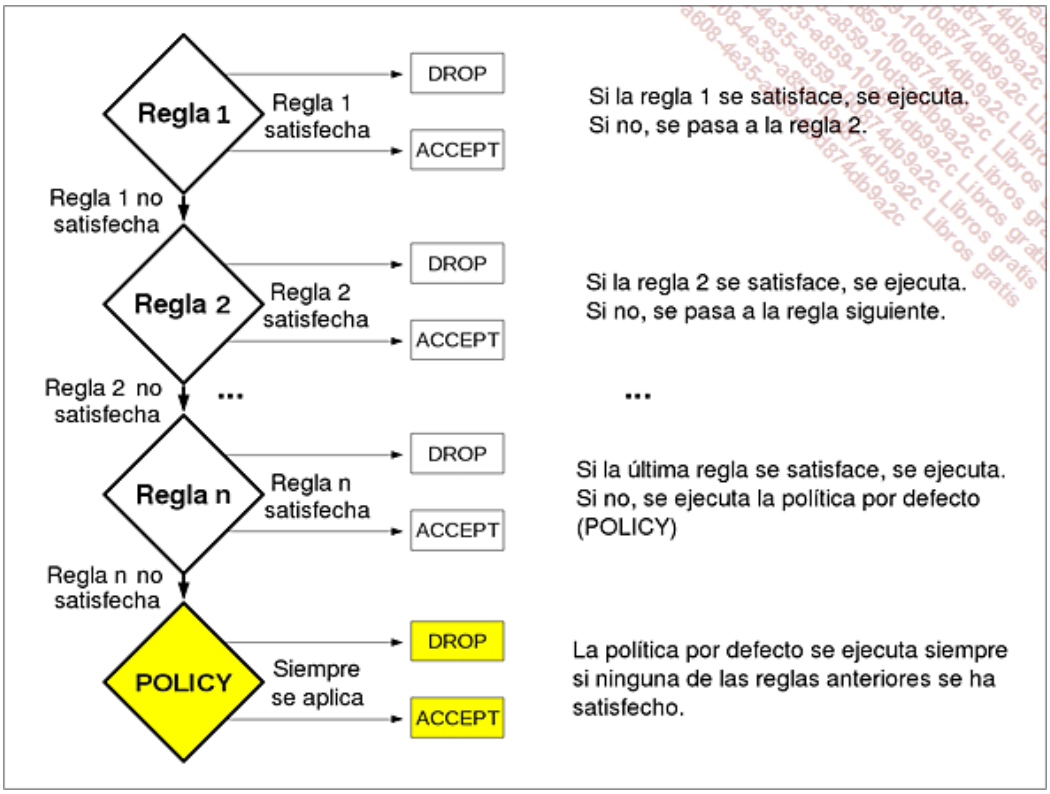
c. Acciones

Cuando se satisface una regla, el sistema genera una acción sobre el paquete comprobado. Las acciones principales son **ACCEPT** que permite el paso del paquete y **DROP**, que lo destruye.

En la sintaxis de iptables, la acción (**target** en el manual en línea) se anuncia con el parámetro **-j**.

Las acciones siempre se escriben en mayúsculas según la sintaxis de iptables.

d. Tratamiento de reglas



Se aplican las reglas una por una para cada paquete filtrado. Si se satisface una regla, se genera una acción sobre el paquete y finaliza su tratamiento. Si no se satisface, se comprueba la siguiente regla. En el caso que ninguna regla se haya satisfecho, el paquete recibe un tratamiento por defecto configurado con una regla específica llamada "política" (policy).

Se pueden mostrar las reglas aplicadas en orden para cada una de las cadenas.

Visualización de reglas efectivas

```
iptables -L
```

Ejemplo de visualización de reglas

Este ejemplo muestra las reglas activas en un sistema Linux sin configurar. Se ve la política aplicada para cada una de las cadenas y se comprueba la ausencia de reglas de filtrado.

```
alfa:~# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
```

El comando **iptables -L** muestra la interpretación de las reglas activas. Si se desea saber quién tiene permisos para establecer estas reglas, se recomienda usar la opción **-S**.

Ejemplo de visualización de reglas

```
Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

según la sintaxis

*La opción **-S** es particularmente útil cuando se enfrenta a un sistema configurado por otra persona y no sabe qué comandos se han introducido para llegar a esa configuración.*

```
alfa:~# iptables -S
-P INPUT ACCEPT
-P FORWARD ACCEPT
-P OUTPUT ACCEPT
alfa:~#
```

Administración de un cortafuegos con iptables

1. Políticas

a. Fundamentos de las políticas de un cortafuegos

Un cortafuegos puede funcionar según dos modelos distintos: "todo lo que no está autorizado está prohibido" o "todo lo que no está prohibido está autorizado". Para definir el comportamiento por defecto, iptables permite definir para cada cadena una acción por defecto.

Definición de la política por defecto de iptables

```
iptables -P cadena acción
```

Donde *cadena* representa el tipo de tráfico (INPUT, OUTPUT y FORWARD), y *acción* el comportamiento deseado (DROP o ACCEPT).

Ejemplo de definición de política

En este ejemplo, se prohíbe todo el tráfico saliente del host aplicando una política de descarte de paquetes salientes.

```
root@test:~$ ping -c 1 192.168.0.10
PING 192.168.0.10 (192.168.0.10) 56(84) bytes of data.
64 bytes from 192.168.0.10: icmp_seq=1 ttl=64 time=0.880 ms
--- 192.168.0.10 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.880/0.880/0.880/0.000 ms
root@test:~$ iptables -P OUTPUT DROP
root@test:~$ ping -c 1 192.168.0.10
PING 192.168.0.10 (192.168.0.10) 56(84) bytes of data.
ping: sendmsg: Operation not permitted
--- 192.168.0.10 ping statistics ---
1 packets transmitted, 0 received, 100% packet loss, time 0ms
root@test:~$
```

b. Configuración de una política básica

Si el host que se desea configurar se sabe que se convertirá en un cortafuegos, es probable que todo el tráfico esté prohibido por defecto. Esta configuración común consiste en establecer para las tres cadenas INPUT, OUTPUT y FORWARD una política de descarte de paquetes.

Configuración de una política restrictiva

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

2. Filtrado de paquetes

a. Política y reglas

Después de haber configurado una política que describe el comportamiento básico del cortafuegos, hay que crear las reglas específicas para el tráfico que se desea dejar pasar o prohibir. La filosofía del cortafuegos es: se define el comportamiento general con las políticas y se gestiona caso por caso el comportamiento específico con reglas.

b. Creación de reglas

Para cada elemento de tráfico que debe estar permitido o prohibido, habrá que crear una regla específica.

Sintaxis de creación de una regla de gestión de tráfico

```
iptables -A cadena -s ip_origen -d ip_destino -p protocolo --dport puerto
-j acción
```

iptables: creación de reglas	
-A *cadena*	Se añade una regla en la cadena *cadena* (INPUT, OUTPUT o FORWARD).
-s *ip_origen*	Opcional: la dirección IP origen de donde provienen los paquetes sometidos a la regla. Si la dirección es una dirección de red, hay que especificar la máscara.
-d*ip_destino*	Opcional: la dirección IP destino a la que van los paquetes sometidos a la regla. Si la dirección es una dirección de red, hay que especificar la máscara.
-p *protocolo*	Indica el protocolo utilizado en el paquete sometido a la regla. Valores comunes: udp, tcp, icmp.
--dport*puerto*	Opcional: indica el puerto de destino del paquete sometido a la regla.
-j *acción*	Indica cómo tratar el paquete sometido a la regla (ACCEPT o DROP).

Autorización de ping saliente y entrante

Cada tipo de flujo tiene que ser el objetivo de una regla de iptables.

```
alfa:~# iptables -A OUTPUT -p icmp -j ACCEPT
alfa:~# iptables -A INPUT -p icmp -j ACCEPT
alfa:~#
```

Autorización del tráfico http que pase por la máquina con origen una red determinada

```
alfa:~# iptables -A FORWARD -s 192.168.1.0/24 -p tcp -dport 80 -j ACCEPT
alfa:~#
```

Una configuración errónea en un cortafuegos puede tener consecuencias drásticas. Para comprobar que la configuración está realizada correctamente se recomienda utilizar un escáner de puertos desde una máquina remota. El comando nmap -F seguido de la dirección IP de la máquina protegida permite comprobar muy rápido (Fastmode) que los puertos están adecuadamente bloqueados o abiertos.

c. Gestión de reglas

Las reglas se aplican en su orden de creación y el sistema les asigna automáticamente un número de orden.

Visualización de los números de reglas efectivas

```
iptables -L cadena --line-numbers -n
```

Donde *cadena* representa la cadena de tratamiento (INPUT, OUTPUT o FORWARD). El parámetro -n no es obligatorio, pero acelera notablemente la visualización del resultado ya que no fuerza a que el comando tenga que resolver las direcciones en nombres.

Eliminación de una regla

```
iptables -D cadena número
```

Donde *número* representa el número de la línea obtenido con el comando anterior y *cadena*representa la cadena de tratamiento (INPUT, OUTPUT o FORWARD).

Inserción de una regla

```
iptables -I cadena número condiciones -j acción
```

Donde *condiciones* representa los criterios de selección del paquete sometido a la regla (direcciones IP, puertos y protocolos).

Ejemplo de gestión de reglas

La gestión dinámica de reglas es tan pesada que su uso se ha establecido en un archivo de script que incluye todas las reglas y se recarga por completo después de cada cambio.

```
alfa:~# iptables -L FORWARD --line-numbers -n
Chain FORWARD (policy DROP)
num  target     prot opt source               destination
1    ACCEPT     tcp  --  192.168.1.0/24       0.0.0.0/0           tcp dpt:23
2    ACCEPT     udp  --  192.168.1.0/24       0.0.0.0/0           udp dpt:53
3    ACCEPT     tcp  --  192.168.1.0/24       0.0.0.0/0           tcp dpt:80
alfa:~# iptables -D FORWARD 1
alfa:~# iptables -L FORWARD --line-numbers -n
Chain FORWARD (policy DROP)
num  target     prot opt source               destination
1    ACCEPT     udp  --  192.168.1.0/24       0.0.0.0/0           udp dpt:53
2    ACCEPT     tcp  --  192.168.1.0/24       0.0.0.0/0           tcp dpt:80
alfa:~# iptables -I FORWARD 1 -s 192.168.1.0/24 -p tcp --dport 22 -j ACCEPT
alfa:~# iptables -L FORWARD --line-numbers -n
Chain FORWARD (policy DROP)
num  target     prot opt source               destination
1    ACCEPT     tcp  --  192.168.1.0/24       0.0.0.0/0           tcp dpt:22
2    ACCEPT     udp  --  192.168.1.0/24       0.0.0.0/0           udp dpt:53
3    ACCEPT     tcp  --  192.168.1.0/24       0.0.0.0/0           tcp dpt:80
alfa:~#
```

d. Gestión de los flujos devueltos

En la mayoría de las aplicaciones de red, un host envía un paquete con destino otro host que le responde. Por lo tanto, se establece una comunicación bidireccional. Ahora bien, en la configuración de un cortafuegos, se visualiza perfectamente la comunicación de ida: por ejemplo, desde un navegador a un servidor web a través del puerto 80, en cambio no se ve tan bien las respuestas que se realizan por un puerto aleatorio, por iniciativa del cliente, mayor que 1024.

En los inicios de los cortafuegos, la solución consistía en autorizar cualquier tráfico entrante cuyo puerto era superior al 1024. Los cortafuegos tenían entonces más tendencia a impedir a los usuarios salir antes que evitar las intrusiones en la red.

Pasados unos años, los cortafuegos llamados "stateful" (con estado) son capaces de autorizar dinámicamente los flujos de retorno siempre que sean respuesta a un flujo de salida explícitamente autorizado.

Autorización implícita de flujos de retorno

```
iptables -A cadena -m state --state ESTABLISHED,RELATED -j ACCEPT
```

La opción -m state permite realizar un filtro en función del estado del paquete tratado. Los estados aceptados son ESTABLISHED y RELATED y representan respectivamente paquetes en respuesta a un flujo autorizado y paquetes enviados para una nueva conexión, pero con la iniciativa de una conexión establecida y autorizada (por ejemplo el tráfico de datos ftp relativo al tráfico de comandos ftp).

Ejemplo de configuración completa de un cortafuegos

Se configura en este caso el cortafuegos para que no deje pasar nada, a excepción de las respuestas a cada una de las comunicaciones establecidas, así como los protocolos necesarios para navegar por Internet (http, https y dns).

En este

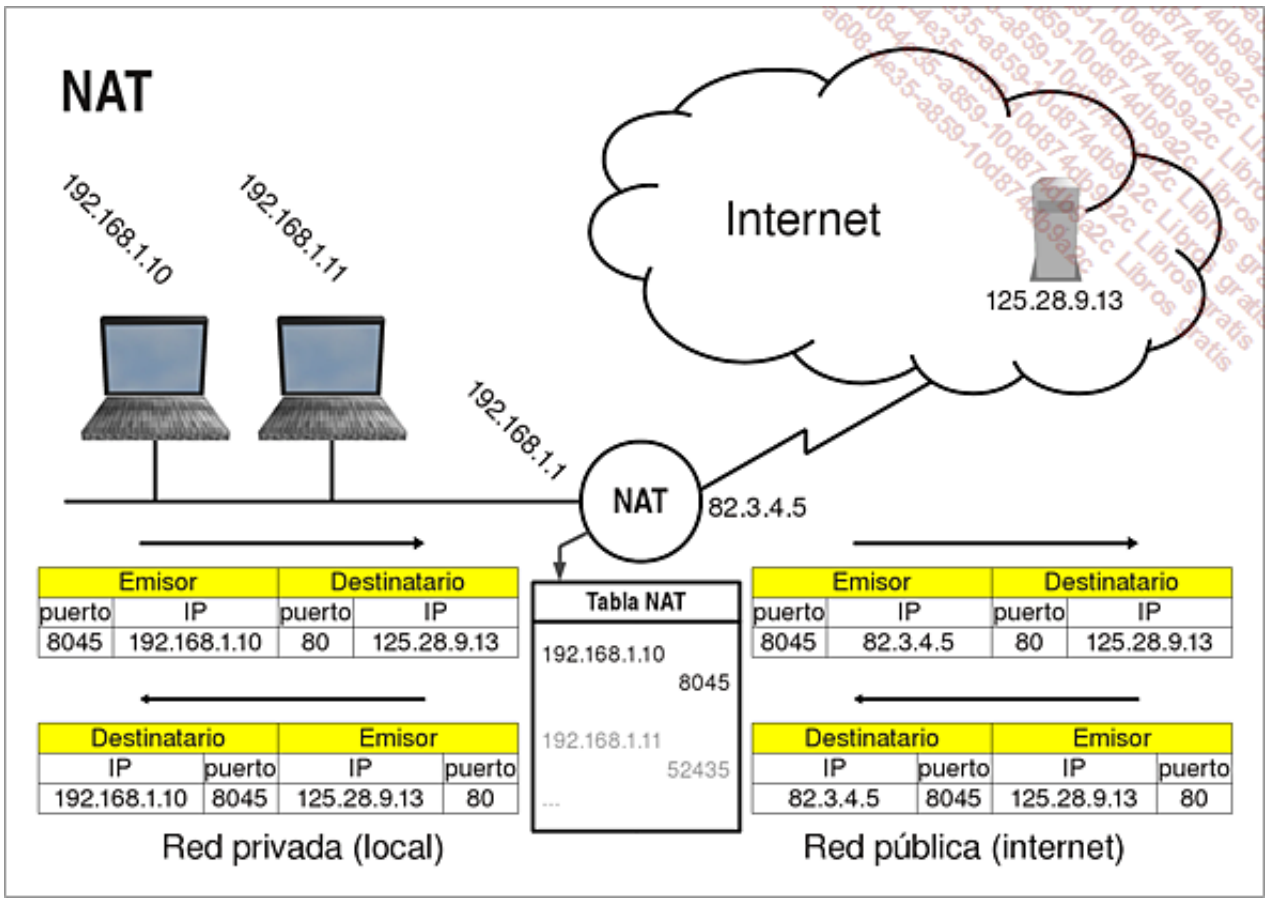
```
alfa:~# iptables -P INPUT DROP
alfa:~# iptables -P OUTPUT DROP
alfa:~# iptables -P FORWARD DROP
alfa:~# iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT
alfa:~# iptables -A FORWARD -s 192.168.1.0/24 -p tcp --dport 80 -j ACCEPT
alfa:~# iptables -A FORWARD -s 192.168.1.0/24 -p tcp --dport 443 -j ACCEPT
alfa:~# iptables -A FORWARD -s 192.168.1.0/24 -p udp --dport 53 -j ACCEPT
```

ejemplo, se configura el cortafuegos para que no deje pasar nada excepto las respuestas a los tráficos establecidos, así como los protocolos necesarios para navegar por Internet (http, https y dns).

La aplicación **fail2ban**, en el caso de que se generen muchos intentos fallidos de conexión a aplicaciones o al propio sistema, permite crear dinámicamente una regla que bloqueará cualquier comunicación del atacante. El conocimiento detallado de su configuración no se exige para la certificación LPI.

3. Gestión de NAT

a. Recordatorio del principio de NAT



NAT consiste en reescribir la cabecera IP de un paquete que viaja de una red pública a una red privada y viceversa.

Las direcciones IP privadas no se pueden enrutar por Internet. Un paquete proveniente de una dirección privada no podría encontrar una ruta de retorno, porque ningún router aceptaría devolverlo a su origen. De todos modos, las redes privadas se usan en todas partes (hay millones de redes 192.168.1.0), sería imposible mantener en las tablas de enrutamiento de los routers de Internet una ruta coherente con la red de origen.

La solución para salir de una red privada consiste en reemplazar la dirección IP privada del emisor por la dirección IP pública (único tipo de dirección en Internet) del router realizando NAT. La trazabilidad de las traducciones (reemplazo de direcciones IP privadas) se realiza gracias al puerto emisor utilizado: para cada

traducción realizada, el router se guarda en memoria el puerto emisor empleado. Como el paquete de respuesta llega al router con la dirección pública del mismo y al mismo puerto que usó en la emisión, la dirección original del cliente se averigua fácilmente por parte del router NAT.

b. Diagnóstico de la configuración NAT de un router

NAT se gestiona en una tabla específica llamada **NAT**. Cualquier configuración vinculada con NAT se realiza con el comando **iptables** especificando que se trabaja en la tabla NAT. Las cadenas que se tratan en la tabla NAT son **PREROUTING**, **POSTROUTING** y **OUTPUT**, que representan el tráfico que hay que modificar antes del enrutamiento, después del enrutamiento o directamente en la salida de la máquina.

Visualización de la configuración NAT

```
iptables -t nat -L
iptables -t nat -S
```

c. Conexión de una red privada a una red pública

En esta configuración, que también es la más corriente, la dirección IP del emisor de los hosts de la red privada se reemplaza por la dirección pública del router NAT.

Configuración de NAT

```
iptables -t nat -A POSTROUTING -o tarjeta_exterior -j acción_nat
```

NAT con iptables: opciones y parámetros	
-t nat	La regla afecta a la tabla NAT.
-A POSTROUTING	Se añade una regla a la cadena POSTROUTING, para el procesado después del enrutamiento.
-o*tarjeta_exterior*	Identifica la tarjeta de red por la cual salen los paquetes del cortafuegos.
-j *acción_nat*	Identifica el modo de acción de NAT, soporta dos opciones: SNAT si la dirección pública es fija y MASQUERADE si la dirección pública es dinámica.

Ejemplo de configuración de NAT

En este ejemplo, eth1 es la interfaz conectada a la red pública.

```
alfa:~# iptables -t nat -A POSTROUTING -o eth1 -j MASQUERADE
alfa:~#
```

4. Scripts de configuración de reglas de filtrado

a. Red Hat e iptables

Los sistemas Red Hat y sus derivados ofrecen un servicio iptables que permite aplicar una configuración de filtrado o NAT automáticamente. El arranque del servicio aplica la configuración y su parada anula todos los filtros. Este funcionamiento es extremadamente práctico y permite gestionar un cortafuegos RedHat cómodamente.

b. Creación de un servicio personalizado de cortafuegos con iptables

Se comprueba bastante rápido que la creación de reglas de filtrado y de NAT con iptables tiene aspectos pesados. Por consiguiente, después de haber determinado qué reglas se necesitan, es interesante escribirlas en un script.

Ejemplo de script de configuración de cortafuegos

Este tipo de script no fuerza a que se tengan que gestionar las reglas una por una en el caso de que se modifique la configuración. Es mucho más fácil insertar una línea en el script que desplazar la numeración de las reglas en memoria. Sin embargo, hay que anular cualquier regla antes de cada ejecución del script.

Por

```
#!/bin/bash
# nombre del archivo: /etc/cortafuegos_on
# Política básica
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP
# NAT con eth0 como interna y eth1 como externa - dirección IP pública fija
iptables -t nat -A POSTROUTING  -o eth1 -j SNAT --to-source 81.2.3.4
# gestión de paquetes devueltos
iptables -A FORWARD -i eth1 -o eth0 -m state --state ESTABLISHED,RELATED -j ACCEPT
# tráfico saliente autorizado
iptables -A FORWARD -i eth0 -o eth1 -p tcp --dport 80 -j ACCEPT
iptables -A FORWARD -i eth0 -o eth1 -p tcp --dport 443 -j ACCEPT
iptables -A FORWARD -i eth0 -o eth1 -p udp --dport 53 -j ACCEPT
```

supuesto, no hay que olvidar convertirlo en ejecutable.

También será útil crear un script de anulación de todas las reglas de filtrado. En efecto, puede ser útil autorizar más o menos provisionalmente todo el tráfico, para una actualización del cortafuegos o para usar una aplicación puntual.

Ejemplo de script de anulación de filtrado

```
#!/bin/bash
# nombre del archivo: cortafuegos_off
# Borrado de reglas
iptables -F
# Política permisiva
iptables -P INPUT ACCEPT
iptables -P OUTPUT ACCEPT
iptables -P FORWARD ACCEPT
```

Finalmente, se puede crear un script de gestión de servicio estándar.

Ejemplo de script de servicio del cortafuegos

Naturalmente, hay que poner este script en el directorio /etc/init.d.

```
#!/bin/bash
# nombre del archivo: cortafuegos
case $1 in
start)
  /etc/cortafuegos_on
  ;;
stop)
  /etc/cortafuegos_off
  ;;
status)
  iptables -L
  ;;
*)
  echo "Syntaxis: /etc/init.d/cortafuegos start|stop|status
  ;;
esac
```

Detección de intrusiones y de vulnerabilidades

1. Sistemas IDS

a. Limitaciones de los cortafuegos

Los cortafuegos, en su modo tradicional de funcionamiento, filtran los paquetes según los valores albergados en las cabeceras de las capas de red o de transporte -y, por tanto, según las direcciones IP o los puertos usados-. Para evitar la protección que aportan los cortafuegos, muchas aplicaciones usan puertos comunes (especialmente el 80 tcp) para lograr transmitir su propio tráfico. Los cortafuegos, a menudo configurados para que dejen pasar los datos por estos puertos comunes, no detectan el peligro.

Para proporcionar un mejor control hay que utilizar un equipamiento más elaborado, capaz de examinar y analizar el tráfico a nivel de aplicación, de forma directa y sin caer en este engaño del puerto erróneo. Estos sistemas se llaman Sistemas de Detección de Intrusos, IDS (*Intrusion Detection System*) en inglés.

b. Técnicas de análisis

Para identificar tráfico perjudicial, los IDS usan tres técnicas: la detección de anomalías, el análisis de protocolos y el análisis de firmas.

La detección de anomalías tiene como objetivo detectar un comportamiento anormal, como por ejemplo un volumen de tráfico ICMP desmesurado, que indicaría que es el blanco o el emisor de un ataque por denegación de servicio.

El análisis de protocolos no busca detectar una acción realmente perjudicial, sino más bien un tráfico de aplicación que no cumple el estándar del protocolo al pie de la letra. Es como la historia del atracador de bancos que fue detenido tontamente porque sus neumáticos eran lisos.

Finalmente, el análisis de firmas permite identificar ataques o comportamientos perjudiciales ya publicados. Es la técnica más eficaz y la que no está sujeta a errores, ya que sólo se generan ataques o intrusiones que ya han tenido éxito en otro lugar y, por tanto, pueden ser debidamente identificados.

c. Fuentes de información

Las técnicas de análisis, que son el análisis de firmas, el análisis de protocolos y la detección de anomalías, se apoyan en información que evoluciona con el tiempo. Es evidente que el análisis de firmas sólo puede realizarse si el IDS conoce la firma del ataque en curso. Además, la naturaleza de las amenazas puede evolucionar. Por ejemplo, un host que enviaba un gran volumen de tráfico SMTP en los años 80 indicaría que el servidor de correo funcionaba bien. La misma situación hoy en día podría demostrar que el host en cuestión está infectado por un caballo de Troya y está enviando un gran volumen de SPAM.

Los IDS obligatoriamente tienen que obtener actualizaciones de sus técnicas de análisis así como de las bases de datos de firmas a intervalos regulares. Los editores de IDS tienen que mantener sistemáticamente sus bases de datos de información al día y los administradores de IDS tienen que descargar regularmente estas bases de datos.

Muchas organizaciones, asociaciones y empresas permiten estar al corriente de las evoluciones en materia de técnicas de intrusión y ataques. Se recomienda conocer la existencia de las principales y, como parte de una administración de red en la que se tiene en cuenta la seguridad, velar tecnológicamente por estas áreas.

Principales organizaciones de alerta y de investigación	
Bugtraq	Lista de difusión dedicada a la publicación de vulnerabilidades, su uso y su corrección.
CERT	*Computer Emergency Response Team*. Esta organización estudia las vulnerabilidades, investiga las evoluciones en términos de redes y seguridad y ofrece servicios relacionados con la seguridad.
CIAC	*Computer Incident Advisory Capability*. Organización de alerta e investigación gestionada por el U.S. Department Of Energy.

2. SNORT

a. Componentes

Snort es el más conocido de los IDS libres. Analiza todo el tráfico y aporta un complemento de seguridad apreciable (incluso indispensable) en la red. Snort se compone de un motor de análisis y de un conjunto de reglas.

Snort se compone de un servicio y de archivos de configuración generalmente ubicados en **/etc/snort**. El principal archivo de configuración es **snort.conf**. Las reglas que se aplicarán se ubican en el subdirectorio **rules**.

Snort también dispone del comando **oinkmaster** para actualizar reglas que tienen su configuración en el archivo **oinkmaster.conf**.

b. Gestión de las fuentes de información

SNORT utiliza archivos de reglas que deben descargarse del sitio web del editor.

Declaración de un archivo de reglas en oinkmaster.conf

```
url = http://www.snort.org/snort-rules/archivo_reglas
```

Donde *archivo_reglas* representa el archivo de reglas en formato tar.gz. Hay que estar suscrito al editor, pero hay otras páginas web que ofrecen archivos de actualización gratuitos. Naturalmente, la calidad del seguimiento depende de los administradores de estos archivos de reglas.

Después de cualquier modificación del archivo de definición de firmas, y a intervalos regulares gracias a una planificación cron, hay que solicitar a snort que cargue sus nuevas reglas. Esta operación se realiza con el comando **oinkmaster**.

Carga de reglas

```
oinkmaster -o dir_reglas
```

Donde *dir_reglas* representa el directorio que contiene las reglas de funcionamiento de snort, suele ser **/etc/snort/rules**. Los archivos de reglas deben llamarse desde el archivo **snort.conf** con el parámetro **include**, como se hace con los parámetros por defecto y las firmas del editor.

c. Gestión de alertas

Cuando Snort detecta tráfico perjudicial, deja una traza en el archivo de registro a través de**syslog** y envía una copia del paquete en un archivo con formato tcpdump (formato libpcap, visualizable con wireshark por ejemplo). También se puede enviar la información a una base de datos (Oracle, MySQL y PostGreSQL son algunas de las bases de datos compatibles).

Ejemplo de declaración de uso de syslog en snort.conf

Esta declaración indica que los elementos deben enviarse al servidor syslog cuya dirección IP es ip_servidor con la categoría "alerta".

```
output alert_syslog: host=ip_servidor, LOG_ALERT
```

3. OpenVAS

OpenVAS (*Open Vulnerability Assessment Scanner*) es una variante del escáner de vulnerabilidades Nessus. Se recomienda conocer su existencia en el marco de la certificación LPI.

a. El servidor OpenVAS

El servidor es el corazón de la suite OpenVAS. Escanea y analiza los hosts de red en busca de

vulnerabilidades conocidas (NVT: *Network Vulnerability Tests*).

b. Clientes OpenVAS

Los clientes OpenVAS son aplicaciones por línea de comandos o con interfaz gráfica que realizan el análisis de los hosts de la red en busca de vulnerabilidades para devolver los resultados al servidor.

c. Obtención de vulnerabilidades

OpenVas ofrece una fuente pública de vulnerabilidades conocidas con el nombre de OpenVas NVT Feed. Permite a los servidores estar al corriente de las últimas vulnerabilidades conocidas y contiene más de 15000 NVT.

Comprobación de los conocimientos adquiridos: preguntas/respuestas

Ponga a prueba sus conocimientos respondiendo a las preguntas siguientes. Estas preguntas no siempre esperan una respuesta cerrada. Las preguntas planteadas en la certificación, a pesar de que abordan los mismos temas, en su mayoría serán planteadas en forma de preguntas de opción múltiple o que requieran una respuesta corta, en palabras escritas en el teclado.

1. Preguntas

1. ¿Un servidor Linux es capaz de enrutar paquetes IP de forma natural?
2. El comando sysctl permite modificar el contenido de ciertos archivos del pseudosistema de archivos /proc. ¿Cómo se construye el parámetro que se le proporciona?
3. ¿Dispone de tabla de enrutamiento un sistema que sólo tiene una tarjeta de red?
4. Si se consulta a iptables con el comando iptables -L, ¿qué tabla de iptables se muestra?
5. Con iptables, ¿cómo se puede aplicar una configuración particular al tráfico con destino el propio sistema distinta de la configuración aplicada al tráfico enrutado por el sistema?
6. En el contexto de uso de iptables, ¿qué sucede si ninguna de las reglas configuradas para una cadena se satisface?
7. ¿En qué aspecto se dice que NAT aporta una protección rudimentaria de las redes privadas?
8. La creación manual de reglas en iptables es muy pesada y no siempre se puede predecir qué se desea filtrar. ¿Cómo se puede automatizar la creación de reglas para bloquear el tráfico molesto?
9. ¿Qué son Bugtraq y CERT?
10. ¿En qué aspecto OpenVAS está bien adaptado para la protección de parques informáticos?

2. Respuestas

1. ¿Un servidor Linux es capaz de enrutar paquetes IP de forma natural?

   Sí, pero esta función siempre está desactivada por defecto. Se puede activar modificando el contenido del archivo /proc/sys/net/ipv4/ip_forward (con el valor 1).

2. El comando sysctl permite modificar el contenido de ciertos archivos del pseudosistema de archivos /proc. ¿Cómo se construye el parámetro que se le proporciona?

   Especificando el archivo que se desea modificar dentro de la estructura de carpetas albergada en /proc/sys. En su ruta hay que reemplazar el separador jerárquico de directorios (la barra) por un punto. El archivo /etc/sysctl.conf se relee en cada arranque mediante el comando sysctl para tener una aplicación permanente de estos parámetros.

3. ¿Dispone de tabla de enrutamiento un sistema que sólo tiene una tarjeta de red?

   Sí, por supuesto. Cualquier sistema IP dispone de su tabla de enrutamiento. Aunque el sistema no sea claramente un router (no está conectado a varias redes), tiene que ser capaz de enrutar paquetes a sus redes de destino. Como mínimo, la tabla de enrutamiento contiene una referencia a la red local y la definición de la ruta por defecto (pasarela por defecto).

4. Si se consulta a iptables con el comando iptables -L, ¿qué tabla de iptables se muestra?

   La tabla filter que, además de estar muy bien, es lo que con mayor frecuencia se desea ver. Sin embargo, este comportamiento puede dar pie a engaños en el aspecto de que muchos administradores incluso ignoran la existencia de la tabla nat que puede consultarse con el comando iptables -t nat -L.

5. Con iptables, ¿cómo se puede aplicar una configuración particular al tráfico con destino el propio sistema distinta de la configuración aplicada al tráfico enrutado por el sistema?

   Para ello, hay que gestionar reglas distintas según la cadena que se desea configurar. La cadena INPUT hace referencia al tráfico con destino el propio sistema, mientras que la cadena FORWARD se aplica al tráfico enrutado a través del sistema.

6. En el contexto de uso de iptables, ¿qué sucede si ninguna de las reglas configuradas para una cadena se satisface?

Se aplica la regla por defecto. Las reglas por defecto se describen en las políticas de iptables (policies), definidas con el parámetro -P. Hay una policy por cadena.

7 ¿En qué aspecto se dice que NAT aporta una protección rudimentaria de las redes privadas?

En el marco de funcionamiento de NAT, las direcciones de las máquinas privadas en la red no van más allá del router NAT (se reemplazan automáticamente por su dirección pública), lo que proporciona una cierta discreción de la red privada. Además, un atacante que quiera penetrar en la red privada desde el exterior no sabrá encontrar la ruta a la red, las direcciones privadas no se pueden enrutar en Internet.

8 La creación manual de reglas en iptables es muy pesada y no siempre se puede predecir qué se desea filtrar. ¿Cómo se puede automatizar la creación de reglas para bloquear el tráfico molesto?

El programa fail2ban tiene como objetivo la creación de reglas dinámicamente, como por ejemplo una regla que prohíba el tráfico proveniente de un usuario remoto que ha hecho tres intentos sin éxito de apertura de sesión SSH.

9 ¿Qué son Bugtraq y CERT?

Organismos de investigación y de vigilancia de la seguridad. Publican los anuncios y los datos técnicos de las vulnerabilidades descubiertas.

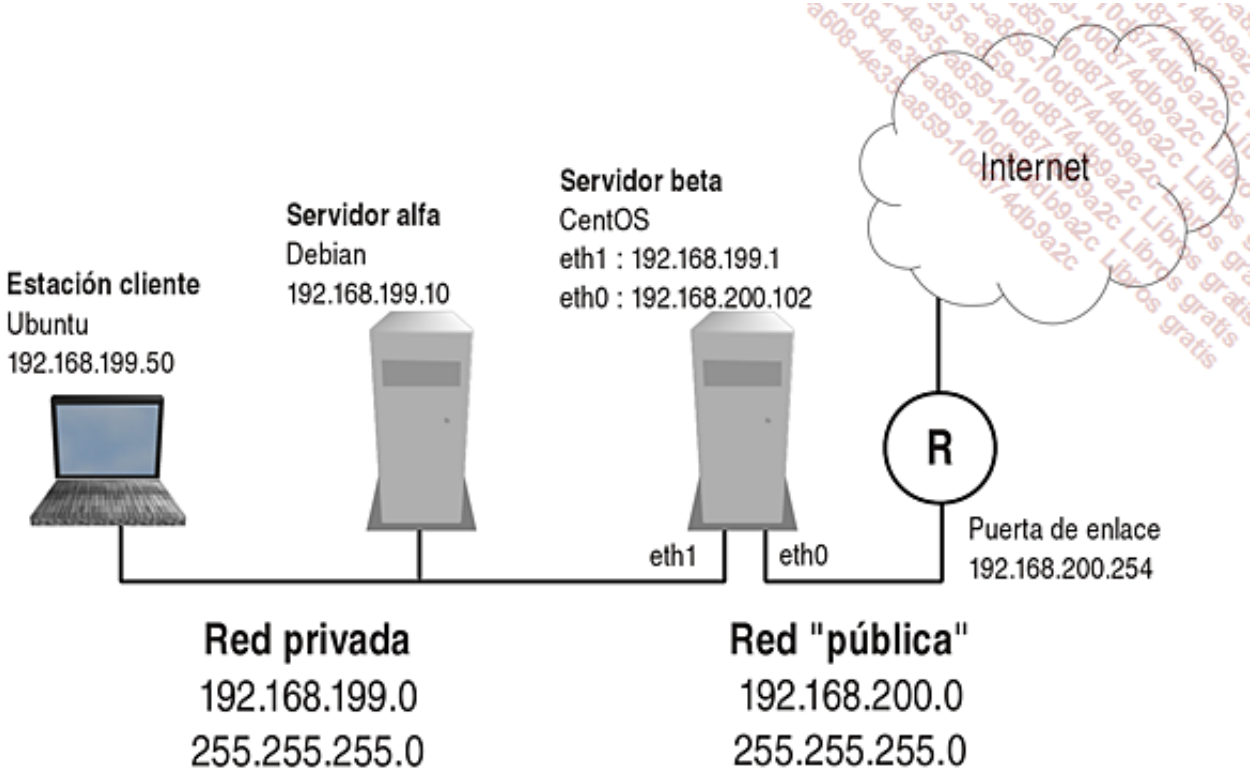
10 ¿En qué aspecto OpenVAS está bien adaptado para la protección de parques informáticos?

Su arquitectura cliente/servidor permite centralizar su configuración y su administración en un servidor; mientras que los componentes cliente se instalan en todas las máquinas del parque que hay que proteger.

Trabajos prácticos

Internet es un mundo hostil. Preocupado por la protección de sus servidores y también por respetar las buenas prácticas en materia de seguridad, decide crear una red privada estrictamente aislada del tráfico protegido por el cortafuegos.

1. Reestructuración de la red local



a. Agregar una interfaz de red al servidor beta

Comandos útiles

- Operaciones relacionadas con el programa de virtualización
- ifconfig
- lspci
- shutdown

Operaciones

1. Detenga el servidor beta con el comando adecuado.
2. Desde la interfaz de gestión de VirtualBox OSE, seleccione el servidor beta y después, en la pestaña **Detalles**, haga clic en **Red**.
3. En la pestaña **Adaptador 2**, haga clic en **Habilitar adaptador de red**. Despliegue a continuación **Conectado a**, elija **Red interna** e informe el campo **Nombre** con el nombre **intnet** que representará una red local privada, accesible solamente para las máquinas virtuales conectadas a esta red privada.
4. Arranque el servidor beta.
5. Compruebe con los comandos apropiados que se ha reconocido una nueva interfaz por parte del sistema.

Resumen de los comandos y resultado por pantalla

Parada del sistema:

```
[root@beta ~]# shutdown -h now
( ... Parada del sistema ... )
```

Comprobación de la interfaz:

b.

```
[root@beta ~]# lspci
00:00.0 Host bridge: Intel Corporation 440FX - 82441FX PMC [Natoma] (rev 02)
00:01.0 ISA bridge: Intel Corporation 82371SB PIIX3 ISA [Natoma/Triton II]
00:01.1 IDE interface: Intel Corporation 82371AB/EB/MB PIIX4 IDE (rev 01)
00:02.0 VGA compatible controller: InnoTek Systemberatung GmbH VirtualBox Graphics
Adapter
00:03.0 Ethernet controller: Advanced Micro Devices [AMD] 79c970 [PCnet32 LANCE] (rev 40)
00:04.0 System peripheral: InnoTek Systemberatung GmbH VirtualBox Guest Service
00:05.0 Multimedia audio controller: Intel Corporation 82801AA AC'97 Audio Controller
(rev 01)
00:07.0 Bridge: Intel Corporation 82371AB/EB/MB PIIX4 ACPI (rev 08)
00:08.0 Ethernet controller: Advanced Micro Devices [AMD] 79c970 [PCnet32 LANCE] (rev 40)
[root@beta ~]#
[root@beta ~]# ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:E4:07:62
          inet adr:192.168.200.102  Bcast:192.168.200.255  Mask:255.255.255.0
          adr inet6: fe80::a00:27ff:fee4:762/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:51 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file txqueuelen:1000
          RX bytes:0 (0.0 b)  TX bytes:6713 (6.5 KiB)

eth1      Link encap:Ethernet  HWaddr 08:00:27:E4:6D:E5
          adr inet6: fe80::a00:27ff:fee4:6de5/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:34 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 b)  TX bytes:6689 (6.5 KiB)

lo        Link encap:Local loopback
          inet adr:127.0.0.1  Mask:255.0.0.0
          adr inet6: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:9846 errors:0 dropped:0 overruns:0 frame:0
          TX packets:9846 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:5485700 (5.2 MiB)  TX bytes:5485700 (5.2 MiB)

[root@beta ~]#
```

Direcciones IP del servidor beta

Comandos y archivos útiles

- /etc/sysconfig/network-scripts/ifcfg-ethx
- ifconfig
- ifup
- route
- vi

Operaciones

1. Encontrar el archivo de configuración de la interfaz eth1.

2. Editarlo e informar los parámetros IP siguientes: 192.168.199.1 255.255.255.0.
3. Activar la interfaz eth1.
4. Comprobar que la configuración se ha cargado correctamente en el sistema.
5. Comprobar que la dirección de la interfaz eth0 se conserva y que la pasarela por defecto no se ha modificado (lo que habría podido pasar si se hubiera informado accidentalmente una puerta de enlace predeterminada en el archivo ifcfg-eth1).

Resumen de los comandos y resultado por pantalla

Archivo /etc/sysconfig/network-script/ifcfg-eth1 modificado:

```
# Advanced Micro Devices [AMD] 79c970 [PCnet32 LANCE]
DEVICE=eth1
BOOTPROTO=none
ONBOOT=yes
HWADDR=08:00:27:e4:6d:e5
IPADDR=192.168.199.1
NETMASK=255.255.255.0
TYPE=Ethernet
```

Activación de la interfaz eth1:

```
[root@beta network-scripts]# ifup eth1
[root@beta network-scripts]#
```

Comprobación de la configuración de eth1:

```
[root@beta network-scripts]# ifconfig eth1
eth1      Link encap:Ethernet  HWaddr 08:00:27:E4:6D:E5
          inet adr:192.168.199.1  Bcast:192.168.199.255  Mask:255.255.255.0
          adr inet6: fe80::a00:27ff:fee4:6de5/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:92 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 b)  TX bytes:15631 (15.2 KiB)

[root@beta network-scripts]#
```

Comprobación de la configuración de la pasarela por defecto y de la interfaz eth0:

c.

```
[root@beta network-scripts]# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 08:00:27:E4:07:62
          inet adr:192.168.200.102  Bcast:192.168.200.255  Mask:255.255.255.0
          adr inet6: fe80::a00:27ff:fee4:762/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:4 errors:0 dropped:0 overruns:0 frame:0
          TX packets:51 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1042 (1.0 KiB)  TX bytes:6713 (6.5 KiB)

[root@beta network-scripts]# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.199.0   0.0.0.0         255.255.255.0   U     0      0        0 eth1
```

```
192.168.200.0   0.0.0.0         255.255.255.0   U     0      0        0 eth0
169.254.0.0     0.0.0.0         255.255.0.0     U     0      0        0 eth1
0.0.0.0         192.168.200.254 0.0.0.0         UG    0      0        0 eth0
[root@beta network-scripts]#
```

Cambios en el cliente para pertenecer a la red privada

Comandos útiles

- Operaciones relacionadas con el programa de virtualización
- Comandos gráficos de gestión de red de la distribución Ubuntu
- ifconfig
- ping

Operaciones

1. En los menús Virtualbox de la estación cliente, desplegar **Dispositivos** y a continuación hacer clic en **Adaptadores de red**.
2. En la pestaña **Adaptador 1**, desplegar **Conectado a** y elegir **Red interna** y seleccionar la red interna **intnet**.
3. En la estación de trabajo Ubuntu, desplegar el menú **Sistema**, después**Preferencias** y, finalmente, elegir **Conexiones de red**.
4. En la ventana **Conexiones de red**, modificar la conexión **Fija eth0** creada anteriormente.
5. En la pestaña **Ajustes de IPv4**, modificar la dirección IP cambiándola por 192.168.199.50 255.255.255.0. Utilizar la pasarela por defecto 192.168.199.1 (servidor beta) y utilizar provisionalmente el servidor DNS de su proveedor de servicios de Internet.
6. Comprobar por línea de comandos la validez de la configuración mediante un ping a la dirección privada del servidor beta (si fuera necesario, reactivar la configuración Fija eth0 haciendo clic en la barra superior - icono de red arriba a la derecha).

Resumen de los comandos y resultado por pantalla

Comprobación de la conectividad:

d.

```
usuario@ubuntu:~$ ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 08:00:27:7b:c8:79
          inet adr:192.168.199.50  Bcast:192.168.199.255  Mask:255.255.255.0
          adr inet6: fe80::a00:27ff:fe7b:c879/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:27984 errors:0 :0 overruns:0 frame:0
          TX packets:92252 errors:5 dropped:0 overruns:0 carrier:5
          collisions:0 txqueuelen:1000
          RX bytes:12348291 (12.3 MB)    TX bytes:9378271 (9.3 MB)

usuario@ubuntu:~$ ping 192.168.199.1
PING 192.168.199.1 (192.168.199.1) 56(84) bytes of data.
64 bytes from 192.168.199.1: icmp_seq=1 ttl=64 time=14.2 ms
64 bytes from 192.168.199.1: icmp_seq=2 ttl=64 time=1.62 ms
64 bytes from 192.168.199.1: icmp_seq=3 ttl=64 time=1.46 ms
^C
--- 192.168.199.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2007ms
rtt min/avg/max/mdev = 1.466/5.789/14.274/6.000 ms
usuario@ubuntu:~$
```

Cambios en el servidor alfa para pertenecer a la red privada

Comandos y archivos útiles

- Operaciones relacionadas con el programa de virtualización
- Archivo /etc/network/interfaces
- ifconfig
- ifup
- ifdown
- ping

Operaciones

1. En los menús de Virtualbox del servidor beta, desplegar **Dispositivos** y después hacer clic en **Adaptadores de red**.
2. En la pestaña **Adaptador 1**, desplegar **Conectado a**, elegir **Red interna** y seleccionar la red interna **intnet**.
3. En el archivo de configuración de red, modificar la dirección IP de la interfaz eth0 con el valor 192.168.199.10 255.255.255.0. Utilizar la pasarela por defecto 192.168.199.1 (servidor beta) y utilizar provisionalmente el servidor DNS de su proveedor de servicios de Internet.
4. Recargar la configuración de la interfaz eth0.
5. Comprobar por línea de comandos la validez de la configuración con un ping a la dirección privada del servidor beta.

Resumen de los comandos y resultado por pantalla

Archivo /etc/network/interfaces modificado:

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug eth0
iface eth0 inet static
address 192.168.199.10
netmask 255.255.255.0
gateway 192.168.199.1

alfa:/etc/network#
```

Comprobación de la conectividad:

2.

```
alfa:/etc/network# ifdown eth0
alfa:/etc/network# ifup eth0
alfa:/etc/network# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 08:00:27:9c:6e:9f
          inet adr:192.168.199.10  Bcast:192.168.199.255  Mask:255.255.255.0
          adr inet6: fe80::a00:27ff:fe9c:6e9f/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:9 errors:0 dropped:0 overruns:0 frame:0
          TX packets:159 errors:0 dropped:0 overruns:0 carrier:0
```

```
          collisions:0 txqueuelen:1000
          RX bytes:714 (714.0 B)  TX bytes:22494 (21.9 KiB)

alpha:/etc/network# ping 192.168.199.1
PING 192.168.199.1 (192.168.199.1) 56(84) bytes of data.
64 bytes from 192.168.199.1: icmp_seq=1 ttl=64 time=0.585 ms
64 bytes from 192.168.199.1: icmp_seq=2 ttl=64 time=0.810 ms
64 bytes from 192.168.199.1: icmp_seq=3 ttl=64 time=1.23 ms
^C
--- 192.168.199.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2006ms
rtt min/avg/max/mdev = 0.585/0.877/1.236/0.269 ms
alfa:/etc/network#
```

Configuración del router y del cortafuegos en el servidor beta

Ahora está tranquilo: su red privada se encuentra protegida trás el servidor beta. Tan protegidos que este servidor sin configurar no deja pasar ningún tipo de tráfico. Deseando poder trabajar un poco, decide gestionar la conectividad entre la red privada e Internet.

a. Configuración de NAT

Comandos y archivos útiles

- /etc/sysctl.conf
- /proc/sys/net/ipv4/ip_forward
- cat
- iptables
- ping
- sysctl

Operaciones

1. Sin usar el comando echo, activar el enrutamiento en el servidor beta.
2. Desde la estación de trabajo, hacer un ping a la interfaz pública de beta.
3. Comprobar que el enrutamiento se ha cargado correctamente consultando el archivo apropiado en el sistema de archivos /proc.
4. Hacer que el enrutamiento se active automáticamente en cada arranque del servidor beta.
5. Comprobar que el servidor beta no realiza NAT: desde la estación de trabajo, hacer un ping a una dirección de la red pública (la pasarela a Internet, por ejemplo).
6. Configurar NAT en el servidor beta.
7. Desde la estación de trabajo, hacer un ping a una dirección de la red pública (la pasarela a Internet, por ejemplo).

Resumen de los comandos y resultado por pantalla

Configuración del enrutamiento en el servidor beta:

Archivo

```
[root@beta ~]# cat /proc/sys/net/ipv4/ip_forward
0
[root@beta ~]# sysctl net.ipv4.ip_forward=1
net.ipv4.ip_forward = 1
[root@beta ~]# cat /proc/sys/net/ipv4/ip_forward
1
[root@beta ~]#
```

/etc/sysctl.conf modificado:

```
# Controls IP packet forwarding
net.ipv4.ip_forward = 1
(...)
```

Configuración de NAT en el servidor beta:

```
[root@beta ~]# iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
[root@beta ~]#
```

Comprobación desde la estación de trabajo:

La

```
usuario@ubuntu:~$ ping 192.168.200.254
PING 192.168.200.254 (192.168.200.254) 56(84) bytes of data.
64 bytes from 192.168.200.254: icmp_seq=1 ttl=63 time=8.45 ms
64 bytes from 192.168.200.254: icmp_seq=2 ttl=63 time=2.82 ms
64 bytes from 192.168.200.254: icmp_seq=3 ttl=63 time=2.43 ms
^C
--- 192.168.200.254 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2006ms
rtt min/avg/max/mdev = 2.434/4.572/8.459/2.753 ms
usuario@ubuntu:~$
```

navegación a Internet también tiene que estar habilitada (puede ser necesario desactivar el uso de un servidor proxy).

b. Política de filtrado estricta

La red local es ahora capaz de navegar libremente por Internet. Sin embargo, en esta fase de la configuración, cualquier protocolo puede circular libremente y esto no se corresponde con sus objetivos. Decide tomar medidas para cambiar esta situación.

Comandos útiles

- iptables
- ping

Operaciones

1. Declarar una política de rechazo para todo tráfico entrante en el servidor beta.
2. Declarar una política de rechazo para todo tráfico saliente del servidor beta.
3. Declarar una política de rechazo para todo tráfico que circule a través del servidor beta.
4. Comprobar la configuración activa.
5. Comprobar que cualquier tipo de tráfico está ahora prohibido.

Resumen de los comandos y resultado por pantalla

Aplicación de las políticas:

Intento

```
[root@beta ~]# iptables -P INPUT DROP
```

de ping desde

```
[root@beta ~]# iptables -P OUTPUT DROP
[root@beta ~]# iptables -P FORWARD DROP
[root@beta ~]#
[root@beta ~]# iptables -L
Chain INPUT (policy DROP)
target     prot opt source               destination

Chain FORWARD (policy DROP)
target     prot opt source               destination

Chain OUTPUT (policy DROP)
target     prot opt source               destination
[root@beta ~]#
```

la estación de trabajo:

c.

```
usuario@ubuntu:~$ ping 192.168.200.254
PING 192.168.200.254 (192.168.200.254) 56(84) bytes of data.
^C
--- 192.168.200.254 ping statistics ---
10 packets transmitted, 0 received, 100% packet loss, time 9011ms

usuario@ubuntu:~$
```

Autorización del tráfico útil

Interesado en alcanzar cierto equilibrio, decide autorizar los protocolos http, https y dns.

Comandos útiles

- iptables

Operaciones

1. Autorizar el tráfico devuelto para cualquier comunicación ya establecida con la cadena FORWARD.
2. Autorizar el tráfico hacia cualquier dirección (la dirección pública en Internet) para el protocolo http (TCP 80).
3. Autorizar el tráfico hacia cualquier dirección (la dirección pública en Internet) para el protocolo https (TCP 443).
4. Autorizar el tráfico hacia cualquier dirección (la dirección pública en Internet) para el protocolo dns cliente/servidor (UDP 53).
5. Comprobar desde la estación cliente que se puede navegar por Internet (no hay que olvidar reconfigurar el navegador para que se conecte directamente a Internet sin pasar por un servidor proxy).
6. Comprobar desde la estación cliente que los pings no logran pasar (en ningún momento se ha autorizado su circulación y la política básica prohíbe cualquier tipo de tráfico no autorizado explícitamente).

Resumen de los comandos y resultado por pantalla

Configuración de las reglas de iptables:

d. Gestión en forma de

```
[root@beta ~]# iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT
[root@beta ~]# iptables -A FORWARD -p tcp --dport 80 -j ACCEPT
[root@beta ~]# iptables -A FORWARD -p tcp --dport 443 -j ACCEPT
[root@beta ~]# iptables -A FORWARD -p udp --dport 53 -j ACCEPT
[root@beta ~]#
```

servicio

Para poder gestionar cómodamente su configuración de filtrado de tráfico, decide crear un servicio que se ejecutará automáticamente en el arranque del sistema.

Comandos útiles

- chmod
- ln
- vi

Operaciones

1. Crear el archivo de script **/opt/scripts/cf0.sh** que anula cualquier forma de filtrado y restablece la política permisiva.
2. Crear el archivo de script **/opt/scripts/cf1.sh** que contiene la política y las reglas de filtrado. Asignarle permisos restrictivos a este archivo para que otros usuarios no puedan ver su contenido.
3. Crear el script de gestión de servicio estándar **cortafuegos**.
4. Crear un enlace **S10cortafuego**s en el directorio correspondiente a su nivel de ejecución por defecto. Este enlace provocará el inicio del servicio en cada arranque del sistema.
5. No olvidar que estos archivos tienen que tener permisos de ejecución.

Resumen de los comandos y resultado por pantalla

Archivo de script ejecutable /opt/scripts/cf0.sh:

Archivo de script

```
#!/bin/bash
iptables -F
iptables -P INPUT ACCEPT
iptables -P OUTPUT ACCEPT
iptables -P FORWARD ACCEPT
```

ejecutable /opt/scripts/cf1.sh:

Archivo

```
#!/bin/bash
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP

iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT

iptables -A FORWARD -p tcp --dport 80 -j ACCEPT
iptables -A FORWARD -p tcp --dport 443 -j ACCEPT
iptables -A FORWARD -p udp --dport 53 -j ACCEPT
```

ejecutable de gestión del servicio /etc/init.d/cortafuegos:

```
#!/bin/bash
case $1 in
start)
  /opt/scripts/cf1.sh
;;
```

```
stop)
  /opt/scripts/cf0.sh
;;
status)
  iptables -L
;;
esac
```

Modificación de los permisos en los archivos de script:

```
[root@beta scripts]# chmod 700 *
[root@beta scripts]# ls -l
total 8
-rwx------ 1 root root 102 aug  8 18:22 cf0.sh
-rwx------ 1 root root 298 aug  8 18:22 cf1.sh
[root@beta scripts]# chmod +x /etc/init.d/cortafuegos
[root@beta scripts]#
```

Creación del enlace simbólico para el nivel de ejecución actual:

```
[root@beta init.d]# runlevel
N 3
[root@beta init.d]# cd /etc/rc3.d
[root@beta rc3.d]# ln -s ../init.d/cortafuegos S10cortafuegos
[root@beta rc3.d]#
```

Requisitos y objetivos

1. Requisitos

Los conocimientos adquiridos con la certificación LPI nivel 1, especialmente:

- ☒ Edición de archivos.
- ☒ Funcionamiento general del servidor X.

2. Objetivos

Al final de este capítulo, será capaz de:

- ☒ Gestionar las autentificaciones SSH.
- ☒ Conocer el funcionamiento de los agentes SSH.
- ☒ Abrir sesiones remotas con SSH.
- ☒ Copiar archivos con scp.
- ☒ Establecer túneles para aplicaciones con SSH.
- ☒ Reenviar sesiones X11 con SSH.
- ☒ Conocer los modos de funcionamiento OpenVPN.
- ☒ Gestionar las autentificaciones OpenVPN con contraseña compartida.
- ☒ Establecer un túnel OpenVPN.

OpenSSH

1. Usos de OpenSSH

Al principio, las sesiones interactivas en sistemas Unix se realizaban con terminales pasivos, que se limitaban a gestionar la entrada/salida, conectados a una unidad central a través del puerto serie. Las pulsaciones del teclado se enviaban sin procesar a la unidad central y la unidad central enviaba como respuesta órdenes de impresión por pantalla. Los ordenadores eran muy caros entonces, el coste relativamente modesto de los terminales pasivos permitía compartir el uso de un ordenador.

Con la generalización de las redes IP y la proliferación de los ordenadores personales, la administración remota de sistemas Unix pasó a realizarse con el protocolo telnet. Se basa exactamente en el mismo principio que los terminales pasivos, excepto en que las pulsaciones de teclado y las órdenes de impresión por pantalla se envían en paquetes transportados por IP. El problema es que la gestión de la seguridad con el protocolo telnet es absolutamente insuficiente: la autentificación se realiza sin encriptar y no hay ningún método de confidencialidad que se aplique a los intercambios entre el cliente y el servidor.

El protocolo SSH está destinado a proporcionar servicios de autentificación y de confidencialidad a los intercambios entre cliente y servidor para el transporte seguro de datos. En la mayoría de los casos simples se utiliza como un "telnet seguro", pero también es capaz de proporcionar un transporte seguro para otros protocolos. La implementación open source del protocolo SSH es "OpenSSH", creada y mantenida por los miembros del proyecto OpenBSD.

2. Gestión de autentificaciones

a. Autentificación por contraseña

El uso más sencillo del cliente SSH, que consiste en abrir una sesión de shell remota de forma segura en redes IP, usa un modo de autentificación simple, que es utilizar una cuenta local del servidor y solicitar al cliente que se autentifique con el nombre y la contraseña de esta cuenta existente en el servidor. La contraseña se verifica a continuación y la autentificación se valida. Sin embargo, esta fase de autentificación por contraseña sirve únicamente para comprobar la validez del cliente. El cliente a su vez puede tener dudas acerca de la identidad del servidor. Dicho de otro modo: ¿se está a punto de hablar con el servidor deseado o con un servidor falso que se aprovechará de los comandos tecleados para obtener información acerca de los sistemas? Para evitar cualquier tipo de riesgo de falsificación de servidor, el cliente realiza una comprobación de la identidad del servidor en la primera conexión. De hecho, se crea una huella digital del servidor y, después de que el cliente valide esta huella, se guarda en un archivo llamado **known_hosts**, alojado en el directorio oculto **.ssh** en el directorio personal del usuario.

Ejemplo de archivo known_hosts

El archivo known_hosts presenta una línea (muy larga) por cada servidor conocido.

```
beta:~# cat .ssh/known_hosts
|1|LPx02U8nHnkSb0czyqVrdXPcW04=|jS0/QdS0HydzPZj8QXxHXC4j6EM= ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAQEAv+kXth0/RSAroNfqeV+IkEMetdWRWYBvbNOqUDDSL/fLylBip9le40xfTe1j
FXuYqAWR+mQMo8Pg37/PUWeetlBCvG4F486UbqUn2Ol5B/1GZqzG7nvbOLcp7CDr6vmqgrk2QZvUZcohWc4L9S6z
zvk3EmQ1AMa+BKo4m+FCG9E1mK4bFtvchVqL1amzGg1jd2QuTzMGNibTdrEi9gSr2TrJ5Se9AhNQkIzZPvrqvVAD
itiggcYNetxaNkPKfW8DdClq+qOVVAQuWnZiO63Mp/0+b+JEutFgNsX8mkt9nx34Yws7s3BnIuT7oU+shxnuy/vj
5But4uUry5tFaTxXCw==
beta:~#
```

b. Autentificación por claves

Sin lugar a dudas, un método más fiable para autentificar conexiones SSH consiste en utilizar claves de autentificación almacenadas localmente en el disco del usuario. La autentificación por claves no exime de la obligación de utilizar una contraseña, pero garantiza al usuario que la máquina remota es con la que se quiere trabajar y no una falsificación.

Creación del par de claves en el cliente

Para que el servidor pueda identificarse formalmente, tiene que tener la clave pública del cliente. Esta clave le permitirá encriptar datos que sólo el cliente propietario de la clave privada asociada podrá descifrar. Por tanto, conviene que se genere en primer lugar esta clave pública en el cliente. Como se trata de criptografía asimétrica, la generación de la clave pública es necesariamente simultánea a la de la clave privada asociada. El comando **ssh-keygen** permite crear estas claves (la pública y la privada).

Generación del par de claves

```
ssh-keygen -t algoritmo
```

Donde *algoritmo* representa el algoritmo usado para la generación de las claves del cliente. Puede tratarse de RSA (versión 1 o 2 de SSH). RSA y DSA son dos algoritmos de encriptación asimétricos que se usan a menudo para autentificar. Si no se especifica un algoritmo, se usa el valor por defecto: RSA.

Generación de un par de claves con los valores por defecto

En este ejemplo se generan un par de claves con el algoritmo por defecto (RSA) para el usuario tata. La representación gráfica (randomart) de la clave no es automática y su impresión por pantalla depende de la versión del comando.

El

```
tata@estacion:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/tata/.ssh/id_rsa):
Created directory '/home/tata/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/tata/.ssh/id_rsa.
Your public key has been saved in /home/tata/.ssh/id_rsa.pub.
The key fingerprint is:
f3:5c:f1:34:6c:1b:a6:4c:5b:c4:6d:30:48:01:76:f4 tata@estacion
The key's randomart image is:
+--[ RSA 2048]----+
|          o+=++o |
|         . ..+..o|
|            o E. |
|           o X + |
|        S   = o  |
|         + .     |
|          o      |
|                 |
|                 |
+-----------------+
tata@estacion:~$
```

comando **ssh-keygen** provoca la creación de dos archivos, creados por defecto en el directorio**.ssh** ubicado directamente en el directorio personal del usuario. Estos dos archivos son por defecto**id_rsa** para la clave privada e **id_rsa.pub** para la clave pública asociada. Aunque no sea obligatorio, se recomienda encarecidamente proteger la clave privada con una contraseña que se solicitará en su creación.

Contenido de los archivos de claves privadas y públicas

A continuación se muestra el contenido de los archivos de claves privadas y públicas. Cabe destacar que los permisos por defecto están limitados en el archivo de clave privado y abiertos en el archivo de clave pública.

En cada

```
tata@estacion:~/.ssh$ ls -l
total 8
-rw------- 1 tata tata 1743 2011-08-09 09:38 id_rsa
-rw-r--r-- 1 tata tata  394 2011-08-09 09:38 id_rsa.pub
tata@estacion:~/.ssh$ cat id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEAs0jrYKKQKiS4f/cCQMhOcc2WTMmGrbXXv3oyz67KUwkm4JumEU1
YkOaNi+WM4nVbkzC7rkUnlXQMxu/EpZLoraNySMHZjUgYiWiRuM4pI0z/atPfjVlwPtGzfUKlqSsP4NCark/9G0
WlMgEXlgpEdeJDmMBRuj98PJjOI/cRGRTgR6JEoevFWMPTDRpoBix3YizVY+dA+unJQPaNKWhoDnCZg7xWi+ZRg
```

```
T2Q1PcbqYKt4xLio+Eei0dvlgu5r5hSvymOdWbXwykywoloIxnzIPiUe7CAxm+KCBA23LQw73pREd1cglS6Gd23
b5Byv/oI6etqs4WOmcJa40Ymvtfbjw== tata@station
tata@estacion:~/.ssh$ cat id_rsa
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: DES-EDE3-CBC,B08C4C3C4B021A76

TzO6ofHOv8sVRDoPj+o7dXfPuXDJaOmQSGhDkWUTC9iGHYnGdHgsig5EKWEez0Zj
YucF9doTpLCv9UsRac6WHRjlQb7AUjk9phEjrKYW4gAfoXNcFY5IiC7fca9i8NQk
YCj4mtzmbJAFc0W9Ax8g0UzZ8bwElIacI28pAdSvVqVHQ6omnVBoWhXhgWTUZaKp
2XbY5gJ7miKW3Y9IPZ3JLukB3j4rTZ0bu8j/UedyXuogpZgYF2vW0GfvtBbfP31F
(...)
RZfBnf+3+KxTvnAtJsMSZc4Glg+9Gch9V+mjU2SfW+T+bUnYLB/6Mpo1aq/akj3r
0G6w12SgjqiOuuXnsCdU8Ox1olCqiHFrk0DyPmwoxcSQygpm2r7FIwL4MPxbELJO
zfk+0wJOmsUANJzeBKd4LXmZykYsAOmf3zZNlS+iU/ZhCBqFmn3/5w==
-----END RSA PRIVATE KEY-----
tata@estacion:~/.ssh$
```

conexión, el servidor mira en el directorio local del usuario que se intenta conectar para ver si existe el directorio **.ssh/authorized_keys** y si contiene la clave pública del cliente. Si éste es el caso, la autentificación del servidor la puede realizar el cliente. Por tanto, el cliente deberá copiar su archivo de clave pública en el directorio **~/.ssh.authorized_keys** del servidor usando algún método de su elección (memoria usb, copia por red).

c. El agente SSH

Para los administradores que necesiten acceder frecuentemente a varias máquinas por SSH, existe un "agente SSH", iniciado con el comando **ssh-agent**, que permite conservar en memoria las claves privadas utilizadas en las autentificaciones. Las claves privadas se transmiten sólo una vez al agente con el comando **ssh-add**. Si se requiere una contraseña de protección de la clave, se solicitará en esta ocasión. A continuación, las claves quedan disponibles sin intervención directa del usuario para cualquier autentificación.

El comando **ssh-add** consulta el directorio **.ssh** en el directorio personal del usuario y busca posibles claves privadas en los archivos **id_rsa**, **id_dsa** e **identity**. Se pueden consultar las claves almacenadas por el agente SSH mediante el comando **ssh-add -l**.

Carga del agente con el comando ssh-agent

El agente nutre variables durante su funcionamiento que permiten gestionarlo más fácilmente.

```
tata@estacion:~$ ssh-agent
SSH_AUTH_SOCK=/tmp/ssh-sRuvox4519/agent.4519; export SSH_AUTH_SOCK;
SSH_AGENT_PID=4520; export SSH_AGENT_PID;
echo Agent pid 4520;
tata@estacion:~$
```

Carga de claves por el agente SSH

El comando ssh-add sin argumentos permite que el agente SSH, que debe haberse iniciado previamente, cargue las claves.

ssh-agent: variables comunes	
SSH_AGENT_PID	El pid del agente que se encuentra en ejecución.
SSH_AUTH_SOCK	El socket creado por el proceso.

```
tata@estacion:~$ ssh-add
Enter passphrase for /home/tata/.ssh/id_rsa:
Identity added: /home/tata/.ssh/id_rsa (/home/tata/.ssh/id_rsa)
tata@estacion:~$
```

Visualización de las claves privadas que el ssh-agent tiene almacenadas

El comando ssh-add -l permite comprobar que el agente ha cargado las claves correctamente.

```
tata@estacion:~$ ssh-add -l
2048 f3:5c:f1:34:6c:1b:a6:4c:5b:c4:6d:30:48:01:76:f4 tata@estacion (RSA)
2048 f3:5c:f1:34:6c:1b:a6:4c:5b:c4:6d:30:48:01:76:f4 /home/tata/.ssh/id_rsa (RSA)
tata@estacion:~$
```

El agente SSH es ante todo una solución de gestión de claves y no tiene como finalidad la creación de claves SSH. El agente SSH sólo puede trabajar con claves ya creadas con el comando ssh-keygen.

3. Confidencialidad en las comunicaciones

a. Sesión interactiva con SSH

La sesión interactiva se abre desde un cliente a un servidor con una cuenta de usuario existente en el servidor.

Apertura de sesión interactiva con SSH

```
ssh usuario@dirección_servidor
```

Sesión interactiva con SSH: opción y parámetros	
usuario	La cuenta de usuario existente en el servidor con la que se realiza la conexión.
dirección_servidor	La dirección IP del servidor con el que se realiza la conexión.

Ejemplo de apertura de sesión interactiva con SSH

```
alfa:~# hostname ; whoami
alfa
root
alfa:~# ssh toto@192.168.0.11
toto@192.168.0.11's password:

toto@beta:~$ hostname ; whoami
beta
toto
toto@beta:~$
```

b. Copia de archivos con SSH

El comando **scp** requiere el daemon SSH y permite copiar archivos de forma segura utilizando los servicios de autentificación y de confidencialidad proporcionados por SSH. La copia puede realizarse del cliente al servidos o viceversa.

Copia de archivos del cliente al servidor con scp

```
scp archivo_local usuario@dirección_servidor:archivo_remoto
```

Copia de archivos del servidor al cliente con scp

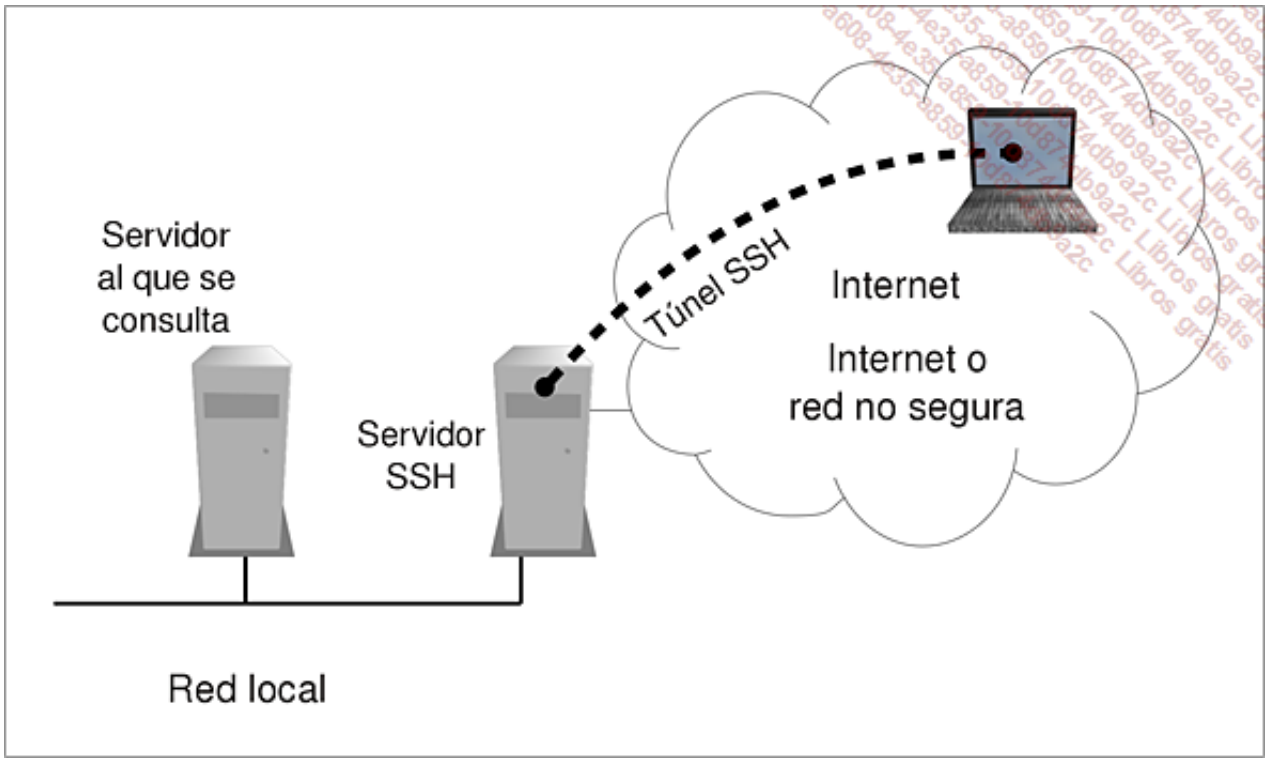
```
scp usuario@dirección_servidor:archivo_remoto archivo_local
```

Copia de archivos con scp: opciones y parámetros	
archivo_local	Ruta relativa o absoluta del archivo local que debe copiarse.

archivo_remoto	Ruta absoluta del archivo remoto que debe copiarse.
usuario	Cuenta de usuario existente en el servidor utilizada para realizar la copia.
dirección_servidor	Dirección IP del servidor que alberga el servicio SSH.

c. Utilización de aplicaciones en túneles SSH

La creación de un túnel SSH permite asegurar las comunicaciones cliente/servidor para protocolos a priori poco seguros. Se establece un túnel desde el cliente al servidor y todo el tráfico entre estas dos máquinas es seguro. El servidor genera entonces otra comunicación no segura con la máquina objetivo del tráfico. Las conexiones de los clientes que deseen utilizar el túnel se crean a través del cliente SSH.



Creación de un túnel SSH

```
ssh -L puerto:destino_tráfico:puerto_destino usuario@servidor
```

Túnel SSH: opciones y parámetros	
-L	Reenvía un puerto local a un servidor SSH (establecimiento del túnel).
puerto	El puerto local que se reenviará.
destino_tráfico	Dirección IP o nombre de la máquina destino del tráfico.
puerto_destino	Puerto al que se reenviará el tráfico en la máquina destino.
usuario	Cuenta de usuario en el servidor usada para desplegar el túnel.
servidor	Dirección IP o nombre del servidor que será el extremo del túnel.

Con esta instrucción, se crea un túnel entre un cliente y un servidor. En el cliente, el tráfico con destino el puerto local se reenvía a través del túnel SSH a la máquina destino en el puerto destino.

d. Reenvío de sesiones X11 con SSH

El servidor X no tiene de forma nativa una fuerte seguridad para sus comunicaciones cliente/servidor. Un uso habitual de SSH consiste en transmitir a través de un túnel SSH aplicaciones gráficas. Para ello, hay que autorizar que el servidor SSH reenvíe este tipo de tráfico y utilizar un cliente compatible con este modo de funcionamiento.

La autorización del reenvío de sesiones X a través de SSH se consigue modificando el archivo de configuración del servidor SSH **/etc/ssh/sshd_config**.

Autorización del reenvío de conexiones X en sshd_config.conf

```
X11Forwarding yes
```

Conexión desde un cliente SSH

```
ssh -X usuario@servidor
```

Donde *usuario* representa la cuenta de usuario utilizada para la conexión y *servidor* la dirección IP o el nombre del servidor al que se conecta. Las aplicaciones gráficas podrán entonces ejecutarse desde la sesión SSH cliente.

OpenVPN

OpenVPN es un programa open source para la creación de túneles seguros (VPN). Contrariamente a las VPN habituales, no se basa en IPsec sino en SSL. Proporciona servicios de autentificación, de confidencialidad y de control de la integridad.

1. Modos de funcionamiento OpenVPN

La certificación LPI no exige profundos conocimientos de OpenVPN, sin embargo hay que conocer lo esencial de sus modos funcionales.

a. Autentificación

Los extremos del túnel, es decir, las dos máquinas que encriptan los flujos de datos salientes y desencriptan los entrantes, tienen que estar mutuamente autentificados. No debería haber ninguna duda acerca de la autenticidad del otro extremo. OpenVPN soporta varios modos de autentificación, pero los dos más comunes son la autentificación por compartición de claves y la autentificación con certificados digitales X509. La primera solución es mucho más sencilla de aplicar, pero también es menos segura. La segunda, si bien es la recomendada, es mucho más difícil de desplegar si no se tiene profundos conocimientos de la infraestructura de clave pública que permite generar los certificados. A menudo es mejor tener una solución de claves compartidas que funcione correctamente que una infraestructura de clave pública tambaleante, mal controlada y, por lo tanto, difícil de mantener.

b. Confidencialidad

La confidencialidad de las comunicaciones está garantizada por el uso de la librería OpenSSL. El algoritmo Blowfish es el que proporciona la encriptación de las comunicaciones por defecto, aunque también se usan los algoritmos simétricos habituales (especialmente AES).

c. Funcionamiento de red

El modo de funcionamiento más sencillo y más fácil de aprender es el modo punto a punto, en el que los dos protagonistas de la vpn son los que deben comunicarse de forma segura: son al mismo tiempo los extremos del túnel y los extremos de la comunicación. También se pueden conectar dos redes entre sí en modo sitio a sitio. Dos servidores OpenVPN proporcionan entonces un túnel, pero los extremos del tráfico son las dos redes conectadas. De este modo, los servidores OpenVPN realizan además una función de enrutamiento entre ambas redes. Finalmente, se puede montar una VPN de acceso remoto en la que una máquina se conecta a una red.

OpenVPN puede funcionar en modo puente, en cuyo caso se conectan dos redes remotas como si se hubiera añadido un cable entre los switches, aunque este cable fuera de 200km. Este modo de funcionamiento puede considerarse como anecdótico, el modo enrutado es de lejos el más utilizado.

El nivel de transporte de los paquetes encriptados usa por defecto UDP aunque también se puede usar TCP.

2. Creación de un túnel punto a punto

a. Gestión de la autentificación

El método de autentificación mediante clave compartida requiere la presencia de un archivo de clave en el formato reconocido por OpenVPN. Este archivo tiene que existir en el servidor y en el cliente, y por tanto copiarse mediante algún método seguro (memoria usb, scp). El archivo puede generarse directamente con el comando **openvpn**.

Generación del archivo de clave secreta

```
openvpn --genkey --secret archivo_clave
```

Donde *archivo_clave* representa el archivo que contiene la clave secreta.

Ejemplo de generación de clave

Se genera a continuación una clave secreta que permitirá realizar la autentificación entre las máquinas en cada extremo del túnel.

```
alfa:/etc/openvpn# openvpn --genkey --secret secret.key
alfa:/etc/openvpn# cat secret.key
#
# 2048 bit OpenVPN static key
#
-----BEGIN OpenVPN Static key V1-----
ae11344ce37de44dcce059ecf9fa573f
a2694d5531bc7ed144a12a099c4ef8ce
(...)
1d37552cd4f29ff6b719588056a60777
579cc2aff71bf339f5293bf08f2ce4df
-----END OpenVPN Static key V1-----
alfa:/etc/openvpn#
```

b. Archivos de configuración

Los archivos de configuración se encuentran por defecto en el directorio **/etc/openvpn**. Aunque la práctica recomienda que los archivos se llamen client.conf y server.conf, cualquier archivo con la extensión .conf servirá.

Formato del archivo de configuración OpenVPN

```
remote servidor
dev tun
ifconfig IP_local IP_remota
secret archivo_clave
route red_remota máscara
```

Archivo de configuración Open VPN: directivas comunes	
remote *servidor*	En el cliente únicamente. servidor indica el nombre o la dirección IP del servidor al que se debe conectar la VPN.
dev tun	Crea una encapsulación de tipo túnel (y no la encapsulación ethernet puente).
ifconfig *IP_local IP_remota*	Establece las direcciones locales y remotas de los extremos de la comunicación. Estas direcciones estarán visibles en forma de interfaz virtual en la configuración de red del host.
secret *archivo_clave*	Indica qué archivo contiene la clave compartida, idéntica en ambas máquinas.
route *red_remota máscara*	Parámetro del cliente: indica la dirección de red privada detrás del servidor para que el tráfico con destino esta red se enrute correctamente por la VPN.

Ejemplo de archivos de configuración OpenVPN

Archivo de configuración del lado servidor.

```
alfa:/etc/openvpn# cat server.conf
dev tun
ifconfig 10.8.0.1 10.8.0.2
secret secret.key
```

Archivo de configuración del lado cliente.

```
beta:/etc/openvpn# cat client.conf
remote alfa
dev tun
ifconfig 10.8.0.2 10.8.0.1
secret secret.key
route 192.168.1.0 255.255.255.0
```

c. Despliegue del túnel vpn

Una vez se han creado los archivos en el cliente y el servidor, basta con iniciar en ambas partes el servicio con su script de inicio.

La comprobación del funcionamiento puede hacerse mediante un ping entre las dos direcciones del túnel. Una captura de tramas permitirá observar el tráfico entre ambos extremos a través del puerto UDP/1194 por defecto.

Ejemplo de comprobación de un túnel punto a punto

Se inicia el servicio con su script estándar, se comprueba la existencia de la interfaz virtual y se controla el funcionamiento del túnel con tráfico de un tipo cualquiera.

```
beta:~# ifconfig tun0
tun0: error fetching interface information: Device not found
beta:~# /etc/init.d/openvpn start
Starting virtual private network daemon: client.
beta:~# ifconfig tun0
tun0      Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          inet adr:10.8.0.2  P-t-P:10.8.0.1  Mask:255.255.255.255
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

beta:~# ping 10.8.0.1
PING 10.8.0.1 (10.8.0.1) 56(84) bytes of data.
64 bytes from 10.8.0.1: icmp_seq=1 ttl=64 time=0.864 ms
```

Comprobación de los conocimientos adquiridos: preguntas/respuestas

Ponga a prueba sus conocimientos respondiendo a las preguntas siguientes. Estas preguntas no siempre esperan una respuesta cerrada. Las preguntas planteadas en la certificación, a pesar de que abordan los mismos temas, en su mayoría serán planteadas en forma de preguntas de opción múltiple o que requieran una respuesta corta, en palabras escritas en el teclado.

1. Preguntas

1. Los conceptos de seguridad principales son la autentificación, la confidencialidad y el control de la integridad. El servicio telnet se describe por su ausencia de seguridad, ¿pero dispone sin embargo de algún mecanismo de seguridad?
2. ¿Cómo conserva un cliente SSH el registro de los servidores a los que ya se ha conectado?
3. ¿ssh-keygen se adapta mejor a la creación de claves públicas o privadas?
4. ¿Qué mecanismo permite conservar en memoria las claves privadas utilizadas en las autentificaciones permitiendo de este modo un uso más cómodo?
5. ¿En qué servicio se basa el comando scp en la máquina remota para copiar archivos de forma segura?
6. ¿Cómo se llama el funcionamiento en el que el tráfico se transporta por SSH y, por lo tanto, queda protegido gracias a las funciones nativas de seguridad de este protocolo?
7. ¿Se pueden reenviar sesiones gráficas X11 en un túnel SSH?
8. ¿Qué diferencia hay entre un túnel vpn sitio a sitio y un túnel vpn punto a punto?
9. ¿OpenVPN puede conectar dos máquinas remotas sin proporcionar enrutamiento entre las dos máquinas?
10. ¿Cómo puede un usuario visualizar si un túnel Open VPN se ha montado, a priori, en su máquina?

2. Respuestas

1. Los conceptos de seguridad principales son la autentificación, la confidencialidad y el control de la integridad. El servicio telnet se describe por su ausencia de seguridad, ¿pero dispone sin embargo de algún mecanismo de seguridad?

   Sí, el que se estimaba suficiente en la época de creación del protocolo. Telnet no ofrece ni un control de integridad serio ni una encriptación de datos que proporcione algún tipo de confidencialidad en las comunicaciones. En cambio, el protocolo telnet soporta una autentificación por contraseña. El fallo de esta autentificación es que la transmisión de esta contraseña se realiza sin encriptar, haciendo su intercepción relativamente fácil.

2. ¿Cómo conserva un cliente SSH el registro de los servidores a los que ya se ha conectado?

   Los clientes guardan un registro de cada conexión establecida con servidores SSH y almacenan una huella digital de los servidores en el archivo known_hosts, situado en el directorio oculto .ssh del directorio personal del usuario. Es importante que no haya dudas acerca de la validez del servidor: la encritptación utilizada por SSH permite escaparse de todo intento de lectura realizado con medios razonables, pero es relativamente fácil suplantar la identidad de un servidor usando su nombre y su dirección IP por ejemplo. En esta situación, el usuario tecleería con total confianza comandos que el atacante podría recuperar fácilmente.

3. ¿ssh-keygen se adapta mejor a la creación de claves públicas o privadas?

   La creación de claves públicas y privadas requiere que se haga conjuntamente. Cualquier comando que crea una debe obligatoriamente crear la otra al mismo tiempo. A veces el manual o la documentación presentan una operación antes de la otra, pero en realidad el par de claves se crea a la vez. Es imposible a partir de la clave pública determinar la clave privada asociada y viceversa.

4. ¿Qué mecanismo permite conservar en memoria las claves privadas utilizadas en las autentificaciones permitiendo de este modo un uso más cómodo?

   El comando ssh-agent permite que las claves privadas se almacenen de forma cómoda. Las claves privadas (y las públicas) se crean inicialmente con el comando ssh-keygen y se facilitan al agente con el comando ssh-add. Este agente se carga con el comando ssh-agent. El agente SSH es un programa residente y los programas que

requieran autentificaciones son los clientes.

5 ¿En qué servicio se basa el comando scp en la máquina remota para copiar archivos de forma segura?

El comando scp sólo necesita el servicio SSH en la máquina remota, también utilizado para las sesiones remotas.

6 ¿Cómo se llama el funcionamiento en el que el tráfico se transporta por SSH y, por lo tanto, queda protegido gracias a las funciones nativas de seguridad de este protocolo?

Se llama túnel SSH. Las aplicaciones no requieren modificaciones para este uso, sólo el transporte se ve afectado.

7 ¿Se pueden reenviar sesiones gráficas X11 en un túnel SSH?

Sí, pero para ello hay que autorizar este uso informando la directiva X11Forwarding con el valor yes en el archivo de configuración sshd_config del servidor SSH.

8 ¿Qué diferencia hay entre un túnel vpn sitio a sitio y un túnel vpn punto a punto?

Un túnel punto a punto conecta dos máquinas entre ellas de forma segura. Se garantizan todas las funciones de seguridad (autentificación, confidencialidad, integridad), pero solamente entre ambas máquinas. Usando el modo sitio a sitio, se aplican las mismas funciones al túnel, pero todos los hosts de ambas redes conectadas pueden comunicarse entre ellos mediante el túnel.

9 ¿OpenVPN puede conectar dos máquinas remotas sin proporcionar enrutamiento entre las dos máquinas?

Sí, usando el modo puente en el que el túnel conecta directamente ambas máquinas que se encuentran en la misma subred. Este modo de uso es poco frecuente.

10 ¿Cómo puede un usuario visualizar si un túnel Open VPN se ha montado, a priori, en su máquina?

Consultando la configuración de red con el comando ifconfig. Una interfaz virtual, generalmente llamada tun0, debe mostrarse con la dirección IP asociada a esta interfaz. La presencia de esta interfaz virtual no asegura el correcto funcionamiento del túnel, pero es necesaria para su funcionamiento.

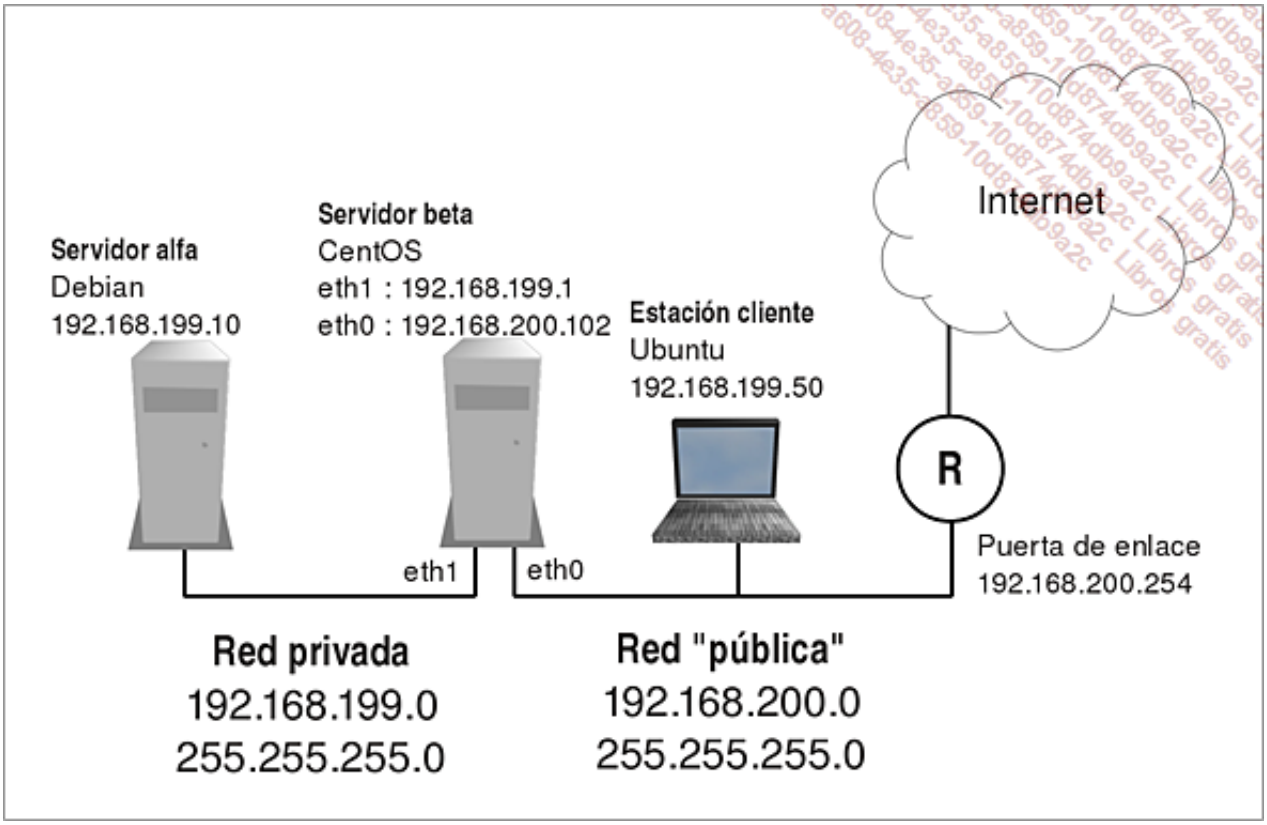
Trabajos prácticos

Aparecen nuevas necesidades. Es necesario acceder desde el exterior a un servidor de intranet situado en el servidor alfa. Duda entre dos soluciones y decide probar ambas.

Estos ejercicios requieren que la red de pruebas se haya reorganizado como se describe en los trabajos prácticos del capítulo Protección de redes.

1. Gestión de la red de pruebas

a. Recolocación de la estación de trabajo



Para la realización de las pruebas se necesita que una estación cliente esté situada en la red pública. Se puede usar una nueva máquina, pero lo más sencillo es desplazar provisionalmente la estación de trabajo Ubuntu a la red pública.

1. En los menús de Virtualbox de la estación cliente, desplegar **Dispositivos** y, a continuación, hacer clic en **Adaptadores de red**.
2. En la pestaña **Adaptador 1**, desplegar el campo **Conectado a** y elegir **Adaptador puente**.
3. En la estación de trabajo Ubuntu, desplegar el menú **Sistema**, seleccionar**Preferencias** y elegir **Conexiones de red**.
4. En la ventana **Conexiones de red**, modificar la conexión **Fija eth0** creada anteriormente.
5. En la pestaña **Ajustes IPv4**, modificar la dirección IP con el valor 192.168.200.50 255.255.255.0 (o una dirección que sea acorde al plan de direccionamiento de la red pública). Modificar también la puerta de enlace.
6. Comprobar por línea de comandos la validez de la configuración haciendo ping a la dirección del servidor beta (192.168.200.102, según el plan de direccionamiento del ejemplo). Si fuera necesario, reactivar la configuración Fija eth0 haciendo clic en la barra de menú superior - icono de red arriba a la derecha.

b. Parada del cortafuegos

Comandos útiles

- Scripts personalizados de gestión del cortafuegos
- iptables

Operaciones

1. Para poder hacer correctamente la práctica sin interferencia por parte del cortafuegos, hay que desactivarlo en el servidor beta. Utilizar para ello los scripts creados en el capítulo Protección de redes.
2. Sólo en caso de necesidad: Si no se dispone de los scripts personalizados, teclear los comandos siguientes:

```
iptables -F
iptables -P INPUT ACCEPT
iptables -P OUTPUT ACCEPT
iptables -P FORWARD ACCEPT
```

3. Comprobar que se han anulado todos los filtros.

Resumen de los comandos y resultado por pantalla

Utilización de scripts personalizados:

```
[root@beta ~]# service cortafuegos stop
[root@beta ~]# service cortafuegos status
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
[root@beta ~]#
```

Anulación manual del filtrado (si fuera necesario):

c.

```
[root@beta ~]# iptables -F
[root@beta ~]# iptables -P INPUT ACCEPT
[root@beta ~]# iptables -P OUTPUT ACCEPT
[root@beta ~]# iptables -P FORWARD ACCEPT
[root@beta ~]# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
[root@beta ~]#
```

Instalación del servidor de la intranet

Instalar si procede el servidor Apache en el servidor alfa con el comando siguiente:

```
apt-get install apache2
```

2. Creación de un túnel SSH entre la estación de trabajo y el servidor beta

En este modo de funcionamiento, se establece un túnel SSH entre el cliente y el servidor beta. Por lo tanto, todo el tráfico en la red pública está protegido. Una vez se ha establecido el túnel, el cliente se dirige a uno de sus puertos locales y el tráfico se dirige a una máquina destino, más allá del túnel.

a. Gestión de la autentificación

Como el túnel se establece entre la estación cliente pública y el servidor beta, hay que resolver el problema de la autentificación entre ambas máquinas. Preocupado por aplicar la solución más segura, opta por la autentificación mediante claves SSH.

Comandos útiles

- mkdir
- scp
- ssh-key-gen

Operaciones

1. En la estación cliente, crear el par de claves necesario para la autentificación utilizando el algoritmo dsa. Aceptar las rutas y los nombres por defecto. Habrá que proteger la clave privada con una contraseña (passphrase) a elección del usuario.
2. En el servidor beta, crear la estructura de directorios apropiada para almacenar la clave pública del usuario que establece el túnel. El archivo de clave pública debe encontrarse en el directorio .ssh/authorized_keys del directorio personal del usuario que se conecte.
3. Copiar la clave pública generada al directorio adecuado en el servidor.

Resumen de los comandos y resultado por pantalla

Generación de las claves cliente en la estación de trabajo:

```
toto@estacion:~/temp$ ssh-keygen -t dsa
Generating public/private dsa key pair.
Enter file in which to save the key (/home/toto/.ssh/id_dsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/toto/.ssh/id_dsa.
Your public key has been saved in /home/toto/.ssh/id_dsa.pub.
The key fingerprint is:
fd:55:bf:50:a5:53:0e:21:92:0b:84:13:1c:96:63:6c toto@estacion
The key's randomart image is:
+--[ DSA 1024]----+
|     o+*. ... o.o|
|     .E . .. . =.|
|     o o . .  o.o|
|         ..   .o.|
|        S .  .. .|
|           . .. .|
|            .  . |
|                 |
|                 |
+-----------------+
toto@estacion:~/temp$
```

Creación de los directorios necesarios para el servidor beta:

```
[toto@beta ~]$ hostname
beta
[toto@beta ~]$ id
uid=500(toto) gid=500(toto) grupos=500(toto)
```

Copia de la clave pública desde

```
[toto@beta ~]$ mkdir -p .ssh/authorized_keys
[toto@beta ~]$
```

la estación al servidor:

b.

```
toto@estacion:~$ cd .ssh
toto@estacion:~/.ssh$ whoami
toto
toto@estacion:~/.ssh$ hostname
estacion
toto@estacion:~/.ssh$ ls
id_dsa  id_dsa.pub  known_hosts
toto@estacion:~/.ssh$ scp id_dsa.pub toto@192.168.200.102:/home/toto/.ssh/authorized_keys
toto@192.168.200.102's password:
id_dsa.pub                                    100%  597     0.6KB/s   00:00
toto@estacion:~/.ssh$
```

Creación del túnel

Comandos útiles

- ssh

Operaciones

1. Desde la estación de trabajo pública, establecer un túnel al servidor beta redirigiendo el puerto local 1234 al servidor interno alfa al puerto 80. El usuario propietario del túnel será **toto**.

Resumen de los comandos y resultado por pantalla

Establecimiento del túnel:

c.

```
toto@estacion:~$ ssh -L 1234:192.168.199.10:80 toto@192.168.200.102
toto@192.168.200.102's password:
Last login: Mon Aug 8 13:39:04 2011 from 192.168.200.50
[toto@beta ~]$
```

Validación

Comandos útiles

- navegador web
- netstat

Operaciones

1. Desde la estación cliente en el navegador, abrir una sesión web hacia sí misma (localhost) al puerto 1234. La página web por defecto del servidor alfa debería mostrarse. Se han transmitido los datos encriptados desde la estación y el servidor beta.
2. En el servidor beta, comprobar que hay una sesión SSH establecida entre el cliente y el servidor beta y que existe una sesión http entre el servidor beta y el servidor alfa.
3. En el servidor alfa, comprobar que hay una sesión http abierta por el servidor beta (extremo del túnel).

Resumen de los comandos y resultado por pantalla

Comprobación de las sesiones tcp en el servidor beta:

```
[root@beta ~]# netstat -n | head -5
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address             State
tcp        0      0 192.168.199.1:34210     192.168.199.10:80           ESTABLISHED
tcp        0      0 192.168.200.102:22      192.168.200.50:46647        ESTABLISHED
Active UNIX domain sockets (w/o servers)
[root@beta ~]#
```

Comprobación de las sesiones tcp en el servidor alfa:

```
alfa:/var/www# netstat -n | head -5
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address             State
tcp6       0      0 192.168.199.10:80       192.168.199.1:45678     TIME_WAIT
Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags       Type       State         I-Node Path
alfa:/var/www#
```

3.

Creación de un túnel VPN entre la estación de trabajo y el servidor beta

a. Instalación del software

Instale OpenVPN en el cliente Ubuntu con el comando siguiente:

```
sudo apt-get install openvpn
```

OpenVPN no forma parte de los paquetes estándar de la distribución CentOS. La solución propuesta es añadir el paquete EPEL, un proyecto de código abierto que pretende proporcionar a las distribuciones Fedora y CentOS programas orientados a profesionales que no están incluidos por defecto en estas distribuciones. Una solución más sencilla sería realizar las pruebas en las distribuciones Debian o Ubuntu exclusivamente.

b. Gestión de la autentificación

1. En el servidor beta, descargue la versión actual del paquete EPEL llamado **epel-release** en la siguiente dirección según la arquitectura del servidor beta:
   http://download.fedoraproject.org/pub/epel/6/i386
   o http://download.fedoraproject.org/pub/epel/6/x86_64
   Este paquete permitirá configurar los repositorios alternativos EPEL en su servidor CentOS.
2. Instale el paquete epel descargado:

   ```
   rpm -i epel-release-x-y.rpm
   ```

3. Actualice la lista de paquetes disponibles:

   ```
   yum update
   ```

4. Instale finalmente openvpn:

   ```
   yum install openvpn
   ```

Comandos útiles

- openvpn

- scp

Operaciones

1. En el cliente, genere una clave exportable para OpenVPN. Almacene esta clave en el archivo **clave.sec**.
2. Copie el archivo que contiene la clave en el servidor beta.

Resumen de los comandos y resultado por pantalla

Generación de la clave en el cliente:

```
toto@estacion:~$ openvpn --genkey --secret clave.sec
toto@estacion:~$ cat clave.sec
#
# 2048 bit OpenVPN static key
#
-----BEGIN OpenVPN Static key V1-----
5e1daf78432b5217c1be08b151630622
2f3df08093262bd5e8e12dfddb180f9b
1bb06c684d842bacbe9b67bb3fe76830
3e23899306d15f33451028e8e1a7d78a
d6850f6cfe666d710e5a840e00fc3d18
d1328b3474a23441353983a697ff04c5
45a8457f2e085883e4565df8a920a655
a98ee7252e9f9e8b0377a2988a261d4c
38d0e02407ed26003fab943f8dde4399
67d053533c807bede026c0be5efe2fe7
987103e4d864ca4799be62a52b2cb47c
2d1c0e76c468a3b8d69c4662debfbb0d
ea722255a0158451b5d21187d54258d1
9ff4cdbdc8f8dd4553b96a303c866f1d
2b360353c78797110ab8c06fd96e58d3
8b283865278e1629fb2054f67e4f52e9
-----END OpenVPN Static key V1-----
toto@estacion:~$
```

Copia de la clave al servidor:

c.

```
toto@estacion:~$ scp clave.sec toto@192.168.200.102:/home/toto/clave.sec
toto@192.168.200.102's password:
clave.sec                                     100%  636     0.6KB/s   00:00
toto@estacion:~$
```

Configuración del cliente

Comandos y directivas útiles

- dev
- ifconfig
- remote
- route
- secret
- vi

Operaciones

1. En la estación cliente, crear el archivo de configuración **/etc/openvpn/cliente.conf**.
2. En el archivo de configuración, indicar que el servidor remoto es **beta**.
3. Indicar que se desea trabajar en modo túnel.
4. Indicar que la dirección local (lado cliente) será **10.9.9.2**.
5. Indicar que la dirección remota (lado servidor) será **10.9.9.1**.
6. Indicar cuál es el archivo de clave secreta que hay que usar.
7. Indicar que el cliente debe tener acceso a la red privada.

Resumen de los comandos y resultado por pantalla

Archivo /etc/openvpn/cliente.conf en la estación cliente:

d.

```
remote 192.168.200.102
dev tun
ifconfig 10.9.9.2 10.9.9.1
secret /home/toto/clave.sec
route 192.168.199.0 255.255.255.0
```

Configuración del servidor

Comandos útiles

- ifconfig
- vi

Directivas útiles

- dev
- route
- secret

Operaciones

1. En el servidor beta, crear el archivo de configuración **/etc/openvpn/servidor.conf**.
2. En el archivo de configuración, indicar que se desea trabajar en modo túnel.
3. Indicar que su dirección local (lado servidor) será **10.9.9.1**.
4. Indicar que la dirección remota (lado cliente) será **10.9.9.2**.
5. Indicar cuál es el archivo de clave secreta que se usará.
6. Indicar que el cliente debe tener acceso a la red privada.

Resumen de los comandos y resultado por pantalla

Archivo /etc/openvpn/servidor.conf en el servidor:

e.

```
dev tun
ifconfig 10.9.9.1 10.9.9.2
secret /home/toto/clave.sec
```

Comprobación

Comandos útiles

- Navegador web
- ping

Operaciones

1. Iniciar el servicio openvpn en el servidor beta.
2. Iniciar el servicio openvpn en la estación cliente.
3. Visualizar las direcciones ip virtuales añadidas a ambas máquinas.
4. Validar las conexiones con un ping.
5. Desde el navegador web en la estación de trabajo, conectarse usando http al servidor alfa. Comprobar que la página se muestra correctamente.

Resumen de los comandos y resultado por pantalla

Inicio del servicio en el servidor beta:

```
[root@beta openvpn]# service openvpn start
Inicialización de de openvpn :                                [  OK  ]
[root@beta openvpn]# ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:E4:07:62
          inet adr:192.168.200.102  Bcast:192.168.200.255  Mask:255.255.255.0
(...)
eth1      Link encap:Ethernet  HWaddr 08:00:27:E4:6D:E5
          inet adr:192.168.199.1  Bcast:192.168.199.255  Mask:255.255.255.0
(...)
lo        Link encap:Bucle local
          inet adr:127.0.0.1  Mask:255.0.0.0
(...)
tun0      Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          inet adr:10.9.9.1  P-t-P:10.9.9.2  Mask:255.255.255.255
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)

[root@beta openvpn]#
```

Inicio del servicio en la estación cliente:

```
toto@estacion:/etc/openvpn$ sudo /etc/init.d/openvpn start
* Starting virtual private network daemon(s)...
* Autostarting VPN 'client'                                   [ OK ]
toto@estacion:/etc/openvpn$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:7b:c8:79
          inet adr:192.168.200.50  Bcast:192.168.200.255  Mask:255.255.255.0
(...)
lo        Link encap:Bucle local
          inet adr:127.0.0.1  Mask:255.0.0.0
(...)
tun0      Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          inet adr:10.9.9.2  P-t-P:10.9.9.1  Mask:255.255.255.255
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)

toto@estacion:/etc/openvpn$
```

Comprobación desde el cliente:

```
toto@estacion:/etc/openvpn$ ping -c 1 192.168.200.102
PING 192.168.200.102 (192.168.200.102) 56(84) bytes of data.
64 bytes from 192.168.200.102: icmp_seq=1 ttl=64 time=1.03 ms

--- 192.168.200.102 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 1.032/1.032/1.032/0.000 ms
toto@estacion:/etc/openvpn$ ping -c 1 192.168.199.10
PING 192.168.199.10 (192.168.199.10) 56(84) bytes of data.
64 bytes from 192.168.199.10: icmp_seq=1 ttl=63 time=5.40 ms

--- 192.168.199.10 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 5.405/5.405/5.405/0.000 ms
toto@estacion:/etc/openvpn$
```

Cabe destacar que con el túnel OpenVPN se obtienen interfaces virtuales que puede usar cualquier aplicación. Con el túnel SSH, se está estrechamente ligado a la aplicación asociada al túnel.

Requisitos y objetivos

1. Requisitos

Los conocimientos adquiridos con la certificación LPI de nivel 1, especialmente:

- ☒ Edición de archivos.
- ☒ Conocer los formatos de compresión gzip y bzip2.
- ☒ Conocer el formato de archivado cpio.

2. Objetivos

Al final de este capítulo, será capaz de:

- ☒ Conocer el principio de una aplicación compilada.
- ☒ Gestionar librerías de aplicación.
- ☒ Realizar una compilación GNU clásica.
- ☒ Instalar y desinstalar fuentes compiladas.
- ☒ Gestionar módulos del kernel.
- ☒ Parchear una aplicación.
- ☒ Preparar la compilación de un kernel (todos los parámetros por defecto).
- ☒ Compilar un kernel.
- ☒ Integrar un kernel en un sistema existente.

Compilación de aplicaciones

1. Características generales

a. Principios de la compilación

Los programas que se utilizan en informática generalmente pertenecen a dos familias: los programas interpretados y los programas compilados. Un programa interpretado se escribe en un lenguaje de programación (basic, perl, shell, etc.) y hay un programa específico que tiene que leerlo para que se pueda ejecutar llamado intérprete. En cada ejecución, el intérprete tiene que recorrer todo el código del programa. Un programa compilado se escribe en un lenguaje de programación (Pascal, C, C++, etc.) y a continuación se pasa a un compilador. El compilador es un programa ejecutable que lee el código del programa que se compilará (llamado código fuente) y generará con esta operación otro programa ejecutable, en modo binario, que podrá ejecutarse independientemente del compilador. La mayoría de los programas usados en el entorno Linux son compilados y el kernel Linux es un ejemplo particular.

b. ¿Cuándo hay que compilar?

La mayoría de las veces se proporcionan las aplicaciones en forma de paquete ya compilado y listo para su uso. En estas condiciones, la compilación es una operación cuyo responsable es el creador del paquete y el usuario no tiene por qué preocuparse. El éxito de distribuciones como Ubuntu viene en parte de su gran número de paquetes existentes y disponibles bajo demanda.

Sin embargo, puede darse el caso en que uno mismo tenga que compilar una aplicación. Por ejemplo, si se desea tener una versión del software tan reciente que todavía no está disponible en forma de paquete o bien que el paquete no exista en la distribución. Además, la compilación se puede personalizar con opciones, mientras que el creador del paquete ha tomado elecciones arbitrarias en lo que respecta a estas opciones de compilación. En estas condiciones, puede desear compilar uno mismo su aplicación y obtener, de este modo, un funcionamiento específico.

c. Recordatorio sobre las utilidades de descompresión

Las fuentes de los programas utilizados en la compilación de aplicaciones casi siempre están disponibles en forma de archivos comprimidos. Por tanto, hay que recordar la sintaxis que permite descomprimir un archivo en formato tar comprimido, que es, con diferencia, el formato más habitual.

Descompresión de un archivo en formato tar comprimido en gzip

```
tar xzf archivo.tgz
```

Descompresión de un archivo en formato tar comprimido en bzip2

```
tar xjf archivo.tar.bzip2
```

La extensión de los archivos es absolutamente convencional y puede variar.

2. Procedimiento de compilación GNU

En la mayoría de las situaciones, la compilación es una operación que pertenece al desarrollador: el desarrollador escribe su programa, lo compila y libera el código ejecutable listo para usar. Por tanto, el gran público desconoce las habilidades necesarias para compilar. El mundo open source cambia un poco el concepto: por definición, el código fuente de todos los programas está disponible y puede ser frecuente que el usuario final tenga que compilar él mismo su aplicación. De este modo, se ha definido un procedimiento de compilación estándar para que cualquier usuario pueda realizar esta operación.

a. Obtención de las fuentes

Por definición, el código fuente de una aplicación open source está siempre disponible, generalmente en un sitio web adjunto al proyecto de desarrollo de la aplicación. Particularmente, el sitio web sourceforge.net se

dedica a albergar muchos proyectos de desarrollo.

Una vez que se han descargado las fuentes, basta con extraerlas de su archivo e irse al directorio que se ha extraído. Todas las operaciones relacionadas con la compilación se realizarán desde la raíz de este directorio.

b. Configuración de la compilación

La compilación tiene una serie de requisitos: la existencia del compilador, la posible presencia de librerías necesarias para compilar el programa y, sobre todo, un archivo de respuesta que será leído por el compilador en la compilación. En el procedimiento estándar de compilación GNU, debe haber un script llamado **configure** en el directorio raíz de las fuentes y este script es el que se encarga de realizar estas tres operaciones. El desarrollador del programa es su autor y lo libera con las fuentes.

Durante la ejecución, posiblemente con opciones de compilación, este script comprobará el entorno y devolverá un mensaje de error en el caso que se produzca una ausencia en el entorno de compilación (compilador y librerías necesarias). Si todo va bien, este script acabará generando archivos como resultado (uno por cada subdirectorio existente en el directorio de los fuentes)llamados **Makefile**. Estos archivos se crean a partir de las opciones que se han facilitado al script de configuración y de un archivo plantilla **Makefile.in**. Aunque una curiosidad justificada incita a la lectura del contenido de estos archivos, no es en absoluto necesaria para seguir con las operaciones.

Ejecución del script configure sin opciones

Se ejecuta el script de configuración desde el directorio raíz de los fuentes. Se puede comprobar que ha terminado con la línea "creating Makefile", que constituye el último paso del script.

```
[root@beta rdesktop-1.6.0]# ./configure
checking for gcc... gcc
checking for C compiler default output file name... a.out
checking whether the C compiler works... yes
checking whether we are cross compiling... no
(...)
checking for setmntent... yes
checking build system type... i686-redhat-linux-gnu
checking host system type... i686-redhat-linux-gnu
configure: creating ./config.status
config.status: creating Makefile
[root@beta rdesktop-1.6.0]#
```

Gestión de fallos por el script de configuración

El script de configuración detecta en este ejemplo la ausencia de librerías que son necesarias para el proyecto. Lejos de ser genéricos, se tiene la suerte de que el script devuelve consejos precisos.

```
[root@beta rdesktop-1.6.0]# ./configure
checking for gcc... gcc
checking for C compiler default output file name... a.out
checking whether the C compiler works... yes
checking whether we are cross compiling... no
checking for suffix of executables...
(...)
checking for inttypes.h... yes
checking for stdint.h... yes
checking for unistd.h... yes
checking whether byte ordering is bigendian... no
checking for X... no

ERROR: Could not find X Window System headers/libraries.
Probably you need to install the libx11-dev package.
To specify paths manually, use the options --x-includes and --x-libraries.
[root@beta rdesktop-1.6.0]#
```

c. Personalización de programas compilados

El desarrollador puede indicar opciones de compilación en la redacción de su script de configuración. El archivo **Makefile** se generará entonces en función de las opciones añadidas en la invocación del script **configure**. La lista de opciones está disponible tecleando el comando**./configure --help**.

Ejecución del script configurado con opciones

Para empezar, se ejecuta el script con la opción --help para saber qué opciones hay disponibles. Después, se vuelve a ejecutar con las opciones elegidas.

```
[root@beta rdesktop-1.6.0]# ./configure  --help
`configure' configures rdesktop 1.6.0 to adapt to many kinds of systems.

Usage: ./configure [OPTION]... [VAR=VALUE]...

To assign environment variables (e.g., CC, CFLAGS...), specify them as
VAR=VALUE.  See below for descriptions of some of the useful variables.

Defaults for the options are specified in brackets.

Configuration:
  -h, --help              display this help and exit
      --help=short        display options specific to this package
      --help=recursive    display the short help of all the included packages
  -V, --version           display version information and exit
  -q, --quiet, --silent   do not print `checking...' messages
      --cache-file=FILE   cache test results in FILE [disabled]
  -C, --config-cache      alias for `--cache-file=config.cache'
  -n, --no-create         do not create output files
      --srcdir=DIR        find the sources in DIR [configure dir or `..']
(...)
[root@beta rdesktop-1.6.0]#
[root@beta rdesktop-1.6.0]# ./configure --with-ipv6
checking for gcc... gcc
(...)
configure: creating ./config.status
config.status: creating Makefile
[root@beta rdesktop-1.6.0]#
```

d. Compilación

La compilación se realiza simplemente con el comando **make**, sin parámetros ni opciones desde el directorio raíz de las fuentes donde se encuentran los archivos **Makefile** y **Makefile.in**. Esta operación suele ser bastante larga y desemboca si todo va bien en la generación de archivos binarios compilados. Cabe destacar que, en esta etapa, estos archivos se encuentran exclusivamente en la estructura de directorios de las fuentes.

Compilación mal preparada

Se intenta realizar en este ejemplo una compilación con el comando make sin haber configurado previamente la compilación.

```
[root@beta rdesktop-1.6.0]# make
make: *** No se especificó ningún objetivo y no se encontró ningún makefile.  Alto.
[root@beta rdesktop-1.6.0]#
```

Compilación sin errores

El comando de compilación make ha encontrado sus archivos Makefile

```
[root@beta rdesktop-1.6.0]# make
(...)
gcc -g -O2 -Wall -I/usr/include   -DPACKAGE_NAME=\"rdesktop\"
```

```
-DPACKAGE_TARNAME=\"rdesktop\" -DPACKAGE_VERSION=\"1.6.0\"
-DPACKAGE_STRING=\"rdesktop\ 1.6.0\" -DPACKAGE_BUGREPORT=\"\"
-DSTDC_HEADERS=1 -DHAVE_SYS_TYPES_H=1 -DHAVE_SYS_STAT_H=1
-DHAVE_STDLIB_H=1 -DHAVE_STRING_H=1 -DHAVE_MEMORY_H=1
-DHAVE_STRINGS_H=1 -DHAVE_INTTYPES_H=1 -DHAVE_STDINT_H=1
-DHAVE_UNISTD_H=1 -DL_ENDIAN=1 -DHAVE_SYS_SELECT_H=1
-DHAVE_LOCALE_H=1 -DHAVE_LANGINFO_H=1 -Dssldir=\"/usr\"
-DEGD_SOCKET=\"/var/run/egd-pool\" -DWITH_RDPSND=1 -DRDPSND_OSS=1
-DHAVE_DIRENT_H=1 -DHAVE_DIRFD=1 -DHAVE_DECL_DIRFD=1 -DHAVE_ICONV_H=1
-DHAVE_ICONV=1 -DICONV_CONST= -DHAVE_SYS_VFS_H=1 -DHAVE_SYS_STATVFS_H=1
-DHAVE_SYS_STATFS_H=1 -DHAVE_SYS_PARAM_H=1 -DHAVE_SYS_MOUNT_H=1
-DSTAT_STATVFS=1 -DHAVE_STRUCT_STATVFS_F_NAMEMAX=1 -DHAVE_STRUCT_STATFS_F_NAMELEN=1
-DHAVE_MNTENT_H=1 -DHAVE_SETMNTENT=1
-DKEYMAP_PATH=\"/usr/local/share/rdesktop/keymaps/\" -o rdesktop
rdesktop.o xwin.o xkeymap.o ewmhints.o xclip.o cliprdr.o rdpsnd.o
rdpsnd_dsp.o rdpsnd_oss.o tcp.o iso.o mcs.o secure.o licence.o
rdp.o orders.o bitmap.o cache.o rdp5.o channels.o rdpdr.o serial.o
printer.o disk.o parallel.o printercache.o mppc.o pstcache.o lspci.o
seamless.o ssl.o  -L/usr/lib -lcrypto     -lX11
[root@beta rdesktop-1.6.0]#
```

e. Los objetivos del comando make

El comando make permite realizar la compilación propiamente dicha, pero este mismo comando llamado con algunos argumentos permite realizar diversas acciones relacionadas con la compilación. A estos argumentos se les llama objetivos. Todos los objetivos no siempre están disponibles, y su presencia depende de los objetivos del desarrollador. Sin embargo, los objetivos de instalación de los binarios o de limpieza sencilla de fuentes siempre están disponibles.

f. Instalación de binarios

Desde el directorio de las fuentes, hay que introducir el comando **make install** para desencadenar la instalación de archivos binarios compilados en sus directorios de destino en el interior de la estructura de carpetas del sistema de archivos Linux. La instalación también puede provocar la copia de archivos de manuales o de configuración.

Instalación automática de todos los elementos compilados

El comando make ejecutado con el objetivo install copia los archivos binarios compilados así como todo elemento previsto por el desarrollador. Se requiere tener los permisos adecuados para la escritura en los directorios de destino.

```
[root@beta rdesktop-1.6.0]# make install
mkdir -p /usr/local/bin
/usr/bin/install -c rdesktop /usr/local/bin
/usr/bin/install:
(...)
[root@beta rdesktop-1.6.0]#
```

g. Limpieza de fuentes

El comando **make clean** ejecutado desde el directorio raíz de las fuentes limpia la estructura de carpeta eliminando cualquier elemento ya compilado y permite reiniciar otra compilación a partir de las mismos fuentes y del mismo entorno.

El comando **make mrproper** permite realizar, como su nombre indica, una limpieza completa de cualquier elemento generado localmente, desde los archivos compilados a los archivos de configuración (Makefile) generados previamente.

Limpieza sencilla de fuentes

El comando make ejecutado con el objetivo clean borra todos los elementos generados por la compilación pero conserva los archivos de configuración.

```
[root@beta rdesktop-1.6.0]# ls Makefile
Makefile
[root@beta rdesktop-1.6.0]# make clean
rm -f *.o *~ vnc/*.o vnc/*~ rdesktop rdp2vnc
[root@beta rdesktop-1.6.0]# ls Makefile
Makefile
[root@beta rdesktop-1.6.0]#
```

h. Desinstalación de un programa

El comando **make uninstall** ejecutado desde el directorio raíz de los fuentes limpia el sistema de todos los archivos instalados por el comando **make install**.

Resumen del procedimiento de compilación estándar GNU

```
cd dir_fuentes
./configure
make
make install
```

Donde *dir_fuentes* representa el directorio de las fuentes, obtenido a partir de la extracción del archivo tar comprimido.

3. Entorno de las aplicaciones

a. Librerías

Una librería (*library* en inglés) es un conjunto de elementos preprogramados que pueden utilizar los desarrolladores. De este modo, se puede ganar tiempo gracias a la reutilización de funciones comunes y a que se evita la reescritura de funciones triviales. El uso de librerías en el entorno gráfico también ayuda a proporcionar unidad a los programas con elementos de interfaz coherentes.

La librería libstdc++ está casi siempre disponible en los sistemas Linux ya que la usan muchos programas y las aplicaciones gráficas usan frecuentemente las librerías gtk o qt. Existen centenares de librerías compartidas utilizadas en entornos Linux. Normalmente se encuentran en el directorio **/usr/lib**.

La mayoría de los programas se compilan de forma dinámica (en contraposición a la forma estática). Es decir, se basan en las mismas librerías que las que ha usado el desarrollador, pero están presentes localmente en el sistema. Por lo tanto, las aplicaciones necesitan tener a su disposición las librerías adecuadas durante la ejecución.

Se puede comprobar cuáles son las librerías necesarias para un ejecutable con el comando **ldd**.

Visualización de las librerías utilizadas por un ejecutable

Se puede observar que para cada librería existe el archivo correspondiente en el disco.

El

```
alfa:~# ldd /bin/ls
        linux-gate.so.1 =>  (0xb775b000)
        librt.so.1 => /lib/i686/cmov/librt.so.1 (0xb7744000)
        libselinux.so.1 => /lib/libselinux.so.1 (0xb772b000)
        libacl.so.1 => /lib/libacl.so.1 (0xb7723000)
        libc.so.6 => /lib/i686/cmov/libc.so.6 (0xb75c8000)
        libpthread.so.0 => /lib/i686/cmov/libpthread.so.0 (0xb75af000)
        /lib/ld-linux.so.2 (0xb775c000)
        libdl.so.2 => /lib/i686/cmov/libdl.so.2 (0xb75ab000)
        libattr.so.1 => /lib/libattr.so.1 (0xb75a6000)
alfa:~#
```

comando **ldconfig** permite crear enlaces entre las aplicaciones y las librerías existentes en el sistema. Consulte en su archivo de configuración **/etc/ld.so.conf** cuáles son las rutas que hay que analizar para la búsqueda de librerías. A continuación se generará el archivo **/etc/ld.so.cache** que contiene la lista de

librerías.

Consideración de las librerías locales

```
ldconfig
```

Visualización de las librerías que se pueden usar

```
ldconfig -p
```

Creación del archivo de caché con ldconfig

A continuación se borra la caché para comprobar que el archivo se crea correctamente mediante el comando.

```
root@beta:~$ rm /etc/ld.so.cache
root@beta:~$ ls /etc/ld.so.cache
ls: no se puede acceder a /etc/ld.so.cache: No existe el fichero o el directorio
root@beta:~$ ldconfig
root@beta:~$ ls /etc/ld.so.cache
/etc/ld.so.cache
root@beta:~$
```

Visualización de librerías

Se puede comprobar que el comando ldconfig -p se basa en el archivo de caché que se ha generado anteriormente.

```
[root@beta ~]# ldconfig -p
776 libs found in cache `/etc/ld.so.cache'
        libz.so.1 (libc6) => /usr/lib/libz.so.1
        libz.so (libc6) => /usr/lib/libz.so
        libxslt.so.1 (libc6) => /usr/lib/libxslt.so.1
        libxslt.so (libc6) => /usr/lib/libxslt.so
        libxml2.so.2 (libc6) => /usr/lib/libxml2.so.2
        libxml2.so (libc6) => /usr/lib/libxml2.so
        libxmlsec1.so.1 (libc6) => /usr/lib/libxmlsec1.so.1
        libxmlsec1.so (libc6) => /usr/lib/libxmlsec1.so
        libxklavier.so.11 (libc6) => /usr/lib/libxklavier.so.11
(...)
[root@beta ~]#
```

Para un uso puntual, también se puede informar una ruta de librería en la variable de sistemaLD_LIBRARY_PATH.

Declaración de rutas de librerías

```
LD_LIBRARY_PATH=ruta1:ruta2:...:rutan
export LD_LIBRARY_PATH
```

Donde las *rutas* representan la ruta absoluta del directorio que contiene las librerías.

b. Visualización de llamadas a sistema

Se puede probar el funcionamiento de las aplicaciones visualizando las llamadas a sistema realizadas por la aplicación durante su ejecución. Si se aplica el comando **strace** a un programa, intercepta las llamadas a sistema que han sido realizadas por el proceso así como los signals recibidos por este proceso. Este comando, útil para desarrolladores, tiene que usarse con mucho cuidado en manos no expertas. El comando **ltrace**, funciona de un modo muy parecido pero centrándose en las cargas de librerías e ignora las llamadas a sistema.

Ejemplo de uso del comando strace

Se puede ver la llamada a varias librerías durante la ejecución del comando echo.

```
[root@beta ~]# strace echo hola
execve("/bin/echo", ["echo", "hola"], [/* 35 vars */]) = 0
brk(0)                                  = 0x9d71000
access("/etc/ld.so.preload", R_OK)      = -1 ENOENT (No such file or directory)
open("/etc/ld.so.cache", O_RDONLY)      = 3
fstat64(3, {st_mode=S_IFREG|0644, st_size=60638, ...}) = 0
mmap2(NULL, 60638, PROT_READ, MAP_PRIVATE, 3, 0) = 0xb7fe8000
close(3)                                = 0
open("/lib/libc.so.6", O_RDONLY)        = 3
read(3, "\177ELF\1\1\1\0\0\0\0\0\0\0\0\3\0\3\0\1\0\0\0\340\317\270\0004\
0\0\0"..., 512) = 512
(...)
brk(0)                                  = 0x9d71000
brk(0x9d92000)                          = 0x9d92000
open("/usr/lib/locale/locale-archive", O_RDONLY|O_LARGEFILE) = 3
fstat64(3, {st_mode=S_IFREG|0644, st_size=56464512, ...}) = 0
mmap2(NULL, 2097152, PROT_READ, MAP_PRIVATE, 3, 0) = 0xb7de6000
close(3)                                = 0
fstat64(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(136, 2), ...}) = 0
mmap2(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS,
 -1, 0) = 0xb7ff6000
write(1, "hola\n", 5hola
)                 = 5
close(1)                                = 0
munmap(0xb7ff6000, 4096)                = 0
exit_group(0)                           = ?
[root@beta ~]#
```

Compilación del kernel

Desde el punto de vista de la compilación, el kernel (también llamado núcleo) es casi como una aplicación más, con su código fuente, un procedimiento de compilación y un procedimiento de instalación.

1. Los componentes del kernel

El kernel Linux es el responsable de la gestión del hardware. El concepto de controlador de dispositivo no existe directamente en entornos Linux debido a que los elementos que permiten comunicarse correctamente con un periférico están incluidos en el código del kernel. Uno se acostumbra muy rápido a la comodidad de esta situación: el kernel recién incluido en una distribución Linux puede gestionar directamente el conjunto de dispositivos de un sistema sin tener que instalar controladores adicionales. Por el contrario, el código del kernel tiene la tendencia cada vez más a ser más pesado para gestionar el conjunto de dispositivos existente y su carga total provocaría un consumo de memoria excesivo. Por esta razón, el kernel tiene una estructura modular y sólo se cargan en memoria los módulos necesarios para el correcto funcionamiento del sistema.

a. El corazón del kernel

Se puede llamar "corazón del kernel" a la parte irreductible del kernel, que es la que se carga en su totalidad en memoria. Sólo contiene elementos de los que se está seguro que se necesitarán. El corazón del kernel es un archivo que se encuentra en el directorio /boot y cuyo tamaño es de algunos MB.

b. Módulos

La importancia de los módulos del kernel

Los módulos tienen un papel primordial ya que muchas de las funciones básicas se gestionan en forma de módulos. Si un kernel no dispone de los módulos necesarios para el funcionamiento del sistema, las funciones simplemente no estarán disponibles.

Intento de carga de un recurso no soportado

Este ejemplo se realiza en un sistema cuyo kernel no soporta el formato de sistema de archivos ext3.

```
light:/mnt# mount /dev/hda3 partition
mount: unknown filesystem type 'ext3'
light:/mnt#
```

Los módulos son archivos con la extensión **.ko** que se cargan en memoria en función de las necesidades. Existe a nuestra disposición una serie de comandos que permiten listar los módulos cargados, retirarlos de memoria o cargar otros nuevos.

Los kernels de versiones antiguas (2.4 especialmente) usan archivos de módulos con la extensión ".o".

Manipulaciones puntuales de módulos

Visualización de módulos cargados en memoria

```
lsmod
```

Visualización de los módulos disponibles en el sistema

```
modprobe -l
```

Los archivos correspondientes a estos módulos se encuentran normalmente en el directorio **/lib/modules**, dentro de una estructura de directorios cuyo directorio principal es el nombre del

kernel actual, tal y como se obtiene con el comando **uname -r**.

Quitar un módulo cargado en memoria

```
rmmod nombre_módulo
```

o

```
modprobe -r nombre_módulo
```

Donde *nombre_módulo* representa el nombre del módulo presente en memoria tal y como se muestra con el comando **lsmod**. Ambos comandos (**rmmod** y **modprobe -r**) producen el mismo resultado.

Carga de un módulo en memoria

```
insmod archivo_módulo
```

o

```
modprobe nombre_módulo
```

Donde *nombre_módulo* representa el nombre del módulo tal y como será mostrado por el comando**lsmod** y *archivo_módulo* representa el nombre del archivo de módulo presente en disco. De hecho, el nombre del módulo se obtiene retirando la extensión **.ko** al nombre del archivo.

Carga de un módulo

La carga manual del módulo que faltaba anteriormente hace posible que se pueda montar la partición ext3.

```
light:/mnt# insmod /lib/modules/2.6.26-2-686/kernel/fs/ext3/ext3.ko
light:/mnt# mount /dev/hda3 partition
light:/mnt# mount
/dev/hda1 on / type ext2 (rw,errors=remount-ro)
(...)
/dev/hda3 on /mnt/partition type ext3 (rw)
light:/mnt#
```

Carga forzada de un módulo

Los módulos se cargan en principio durante el arranque en función del hardware que se haya detectado. Sin embargo, se puede forzar la carga de un módulo rellenando un archivo de configuración de módulos. Cualquier módulo mencionado en el archivo **/etc/modules** se cargará por defecto en el arranque.

Configuración de módulos

El archivo **/etc/modules.conf** permite configurar algunos módulos y especialmente definir asociaciones forzadas entre dispositivos y módulos.

Ejemplo de archivo /etc/modules.conf

```
# Asociación forzada del controlador tg3 con la tarjeta de red
alias eth0 tg3
```

A título de comprobación, o para ver si las asociaciones entre el hardware y los módulos han sido realizadas correctamente, se puede mostrar información acerca de los módulos cargados con el comando **modinfo**.

Visualización de la información relacionada con un módulo

Se ve especialmente el archivo .ko que contiene el código del módulo, alguna información relativa al entorno y los alias gestionados dinámicamente por el sistema para el hardware ligado a este módulo.

```
root@servidor:/boot$ modinfo r8169
filename:       /lib/modules/2.6.32-24-generic/kernel/drivers/net/r8169.ko
version:        2.3LK-NAPI
license:        GPL
description:    RealTek RTL-8169 Gigabit Ethernet driver
author:         Realtek and the Linux r8169 crew <netdev@vger.kernel.org>
srcversion:     D37E06388C6313C1D062CC3
alias:          pci:v00000001d00008168sv*sd00002410bc*sc*i*
alias:          pci:v00001737d00001032sv*sd00000024bc*sc*i*
alias:          pci:v000016ECd00000116sv*sd*bc*sc*i*
alias:          pci:v00001259d0000C107sv*sd*bc*sc*i*
alias:          pci:v00001186d00004300sv*sd*bc*sc*i*
alias:          pci:v000010ECd00008169sv*sd*bc*sc*i*
alias:          pci:v000010ECd00008168sv*sd*bc*sc*i*
alias:          pci:v000010ECd00008167sv*sd*bc*sc*i*
alias:          pci:v000010ECd00008136sv*sd*bc*sc*i*
alias:          pci:v000010ECd00008129sv*sd*bc*sc*i*
depends:        mii
vermagic:       2.6.32-24-generic SMP mod_unload modversions
parm:           rx_copybreak:Copy breakpoint for copy-only-tiny-frames (int)
parm:           use_dac:Enable PCI DAC. Unsafe on 32 bit PCI slot. (int)
parm:           debug:Debug verbosity level (0=none, ..., 16=all) (int)
root@servidor:/boot$
```

c. Alrededor del kernel

Ya se ha explicado que el kernel se constituye de una entidad indivisible y de módulos cargados en memoria bajo demanda. En la fase de arranque, el gestor de arranque carga el kernel así como los módulos correspondientes a la configuración hardware del sistema. Para acelerar la fase de detección del hardware y la carga de módulos asociada, la mayoría de sistemas modernos usan un ramdisk (disco virtual cuyo soporte físico es la memoria principal) que contiene el conjunto de módulos. Este ramdisk se genera una vez se ha compilado el kernel y se llama directamente por el gestor de arranque.

d. Gestión de versiones del kernel

El kernel lleva un número de versión de tipo A.B.C, por ejemplo 2.6.15. "A" determina la versión principal del kernel: hasta hace poco la versión 2. "B" representa la versión actual del kernel. Este valor es sistemáticamente par en las versiones estables e impar en las versiones de desarrollo. Finalmente, "C" se incrementa en función de las evoluciones menores del kernel, básicamente correcciones de errores y actualizaciones de nuevo hardware.

En Julio de 2011, el núcleo alcanzó la versión 3.0 teniendo como cambio principal, según su creador Linus Torvalds: ”nada, absolutamente nada”. Este cambio se recibió como una especie de capricho, más o menos para celebrar los veinte años del núcleo Linux. La evolución de esta versión, la más espectacular, para el gran público ha sido el soporte a los periféricos Kinect de Microsoft. Una adición entre tantas otras, soporte de nuevo hardware, correcciones varias y otras evoluciones menores.

Las distribuciones actuales no buscan especialmente proporcionar la última versión disponible del núcleo. El número de funciones implementadas por el núcleo hace que la validación de una versión antes de su distribución sea difícil y compleja. Los sistemas Linux orientados al gran público son más bien vanguardistas e intentan proporcionar las últimas versiones del núcleo, mientras que las destinadas al mundo empresarial premian la estabilidad y liberan núcleos en versiones más antiguas, que de algún modo han superado las pruebas de estabilidad. En junio de 2012, Red Hat ofrecía un núcleo en versión 2.6.32, Ubuntu 3.2.0, mientras que el último núcleo estable disponible en el sitio web de kernel.org estaba en versión 3.4.4.

El comando **uname -r** permite visualizar la versión del kernel en ejecución.

Visualización de la versión del kernel actual

```
toto@servidor:~$ uname -r
2.6.32-24-generic
toto@servidor:~$
```

2. Procedimiento de compilación y de utilización

El procedimiento de compilación siempre debe consultarse en el archivo **README** presente con las fuentes del kernel. Los elementos específicos del kernel están documentados en el directorio **Documentation** proporcionado con las fuentes. El archivo **README** sólo documenta el procedimiento de compilación.

a. Obtención de fuentes

El código fuente del kernel se puede descargar de forma gratuita desde el sitio webhttp://www.kernel.org. Las principales versiones están disponibles. Los enlaces "Full source" permiten descargar el código fuente completo del kernel.

El kernel se libera en forma de archivo tar.bz2, por lo que primeramente hay que descomprimirlo. Como para cualquier compilación de aplicación, la mayor parte del trabajo se realizará en el directorio creado en la extracción del archivo.

Si se trabaja sobre las fuentes del kernel copiadas en la instalación del sistema, el directorio de trabajo debería ser /usr/src/linux. Naturalmente, la documentación se encontrará entonces en /usr/src/linux/Documentation. En caso de trabajar con fuentes nuevas, se recomienda utilizar un directorio neutro.

b. Generación del archivo de configuración

La compilación se realiza en función de la información albergada en el archivo **.config** que se encuentra en la raíz del directorio de fuentes. Este archivo indica para cada elemento del kernel si debe estar presente en el corazón del kernel, presente en forma de módulo o ausente del kernel compilado.

Según el sistema usado, hay varios medios a nuestra disposición para generar este archivo de configuración.

Generación del archivo de configuración: comandos posibles	
make config	Va realizando preguntas al usuario para cada uno de los módulos.
make menuconfig	Presenta una interfaz de texto mejorada.
make xconfig	Presenta una interfaz gráfica.
make gconfig	Presenta una interfaz gráfica.
make defconfig	Genera un archivo de configuración basándose en todos los valores de compilación por defecto.
make oldconfig	Genera un archivo de configuración basándose en un archivo .config ya utilizado para una versión más antigua del kernel.

Aunque la compilación del kernel no presenta ninguna dificultad particular, informar el archivo de configuración requiere una gran capacidad y el conocimiento preciso del hardware.

Ejemplo de creación del archivo de configuración

La compilación detallada del núcleo requiere el conocimiento de todas las tecnologías hardware gestionadas por este núcleo.

Primeras líneas del archivo de

```
[root@beta linux-2.6.34.4]# make config
  HOSTCC  scripts/basic/fixdep
  HOSTCC  scripts/basic/docproc
  HOSTCC  scripts/basic/hash
  HOSTCC  scripts/kconfig/conf.o
(...)
PentiumPro memory ordering errata workaround (X86_PPRO_FENCE) [Y/n/?] y
HPET Timer Support (HPET_TIMER) [Y/n/?] y
Maximum number of CPUs (NR_CPUS) [8] (NEW) 8
SMT (Hyperthreading) scheduler support (SCHED_SMT) [Y/n/?] y
Multi-core scheduler support (SCHED_MC) [Y/n/?] y
Preemption Model
  1. No Forced Preemption (Server) (PREEMPT_NONE)
> 2. Voluntary Kernel Preemption (Desktop) (PREEMPT_VOLUNTARY)
```

```
  3. Preemptible Kernel (Low-Latency Desktop) (PREEMPT)
choice[1-3]: 2
Reroute for broken boot IRQs (X86_REROUTE_FOR_BROKEN_BOOT_IRQS) [N/y/?] (NEW) n
Machine Check / overheating reporting (X86_MCE) [Y/n/?] n
Toshiba Laptop support (TOSHIBA) [M/n/y/?] n
Dell laptop support (I8K) [M/n/y/?] m
Enable X86 board specific fixups for reboot (X86_REBOOTFIXUPS) [N/y/?] n
(...)
CRC-CCITT functions (CRC_CCITT) [M/y/?] m
CRC16 functions (CRC16) [M/y/?] m
CRC calculation for the T10 Data Integrity Field (CRC_T10DIF) [N/m/y/?] (NEW) m
CRC ITU-T V.41 functions (CRC_ITU_T) [M/y/?] m
CRC32 functions (CRC32) [Y/?] y
CRC7 functions (CRC7) [N/m/y/?] (NEW) n
CRC32c (Castagnoli, et al) Cyclic Redundancy-Check (LIBCRC32C) [Y/m/?] y
#
# configuration written to .config
#


[root@beta linux-2.6.34.4]#
```

configuración

Tamaño

```
[root@beta linux-2.6.34.4]# head -15 .config
#
# Automatically generated make config: don't edit
# Linux kernel version: 2.6.34.4
# Mon Aug 8 13:21:04 2011
#
# CONFIG_64BIT is not set
CONFIG_X86_32=y
# CONFIG_X86_64 is not set
CONFIG_X86=y
CONFIG_OUTPUT_FORMAT="elf32-i386"
CONFIG_ARCH_DEFCONFIG="arch/x86/configs/i386_defconfig"
CONFIG_GENERIC_TIME=y
CONFIG_GENERIC_CMOS_UPDATE=y
CONFIG_CLOCKSOURCE_WATCHDOG=y
CONFIG_GENERIC_CLOCKEVENTS=y
[root@beta linux-2.6.34.4]#
```

indicativo del archivo de configuración

La

```
[root@beta linux-2.6.34.4]# wc -l .config
3641 .config
[root@beta linux-2.6.34.4]#
```

configuración de los módulos para una versión de kernel instalada se encuentra en el archivo config-*versión* en el directorio /boot.

Visualización de los archivos de configuración de los kernels

```
root@servidor:/boot$ ls config*
config-2.6.27-11-generic  config-2.6.32-21-generic  config-2.6.32-24-generic
config-2.6.28-16-generic  config-2.6.32-22-generic
config-2.6.31-21-generic  config-2.6.32-23-generic
root@servidor:/boot$ cat config-2.6.32-24-generic
#
# Automatically generated make config: don't edit
# Linux kernel version: 2.6.32-24-generic
```

```
# Thu Aug 19 01:38:31 2010
#
CONFIG_64BIT=y
# CONFIG_X86_32 is not set
CONFIG_X86_64=y
CONFIG_X86=y
CONFIG_OUTPUT_FORMAT="elf64-x86-64"
CONFIG_ARCH_DEFCONFIG="arch/x86/configs/x86_64_defconfig"
CONFIG_GENERIC_TIME=y
CONFIG_GENERIC_CMOS_UPDATE=y
CONFIG_CLOCKSOURCE_WATCHDOG=y
(...)
root@servidor:/boot$
```

c. Compilación del kernel y de los módulos

La compilación se realiza de la forma más trivial, simplemente tecleando el comando **make** desde el directorio raíz de los fuentes. La duración de la operación depende de la potencia de la máquina en la que se realiza, pero una buena hora es a menudo necesaria. Para un kernel con versión 2.6, el comando **make** provoca la compilación del kernel y sus módulos.

Compilación del kernel y de los módulos

Tardará alrededor de una hora o dos...

La

```
[root@beta linux-2.6.34.4]# make
scripts/kconfig/conf -s arch/x86/Kconfig
  CHK     include/linux/version.h
  UPD     include/linux/version.h
  CHK     include/generated/utsrelease.h
  UPD     include/generated/utsrelease.h
  CC      kernel/bounds.s
  GEN     include/generated/bounds.h
  CC      arch/x86/kernel/asm-offsets.s
(...)
  CC      arch/x86/kernel/cpu/cpufreq/speedstep-lib.o
  CC      arch/x86/kernel/cpu/cpufreq/speedstep-smi.o
  LD      arch/x86/kernel/cpu/cpufreq/built-in.o
  CC [M]  arch/x86/kernel/cpu/cpufreq/powernow-k8.o
  CC [M]  arch/x86/kernel/cpu/cpufreq/acpi-cpufreq.o
  CC [M]  arch/x86/kernel/cpu/cpufreq/speedstep-centrino.o
  CC [M]  arch/x86/kernel/cpu/cpufreq/p4-clockmod.o
  CC      arch/x86/kernel/cpu/mcheck/mce.o
  CC      arch/x86/kernel/cpu/mcheck/mce-severity.o
  CC      arch/x86/kernel/cpu/mcheck/mce_intel.o
(...)
```

ejecución del comando **make** provoca la compilación del kernel y de sus módulos. También invoca el comando **depmod** que genera el archivo **modules.dep** de dependencia de módulos.

d. Instalación de módulos

Los módulos se instalan con el comando específico **make modules_install**. Se copian en el directorio **/lib/modules**, en un directorio correspondiente a la versión del kernel.

Visualización de los directorios que contienen los módulos

Cada versión de kernel instalado tiene su directorio de módulos correspondiente.

```
root@servidor:~$ ls /lib/modules/
2.6.27-11-generic  2.6.28-16-generic  2.6.32-22-generic
2.6.27-7-generic   2.6.31-21-generic  2.6.32-23-generic
2.6.27-9-generic   2.6.32-21-generic  2.6.32-24-generic
```

```
root@servidor:~$
```

e. Instalación del kernel

El kernel sin módulos se encuentra en el directorio de fuentes en la ruta relativa **arch/x86/boot**para las versiones de 32 bits o **arch/ia64/boot** para las versiones de 64 bits con el nombre**bzImage**. Su instalación en el sistema en producción se realiza copiando simplemente este archivo en el directorio **/boot**. Se puede usar perfectamente el nombre por defecto (bzImage), pero es recomendable renombrarlo para tener en cuenta la versión compilada.

Las compilaciones que se realicen con versiones antiguas de kernel pueden generar un archivozImage y no un bzImage. El prefijo z o bz indica si el formato de compresión del archivo de kernel es gzip (z) o bzip2 (bz).

Un kernel recién compilado siempre debe instalarse añadiéndose al kernel existente. Nunca reemplace un kernel que funciona por un kernel nuevo.

Instalación del kernel

Las buenas prácticas recomiendan que el archivo de kernel tenga un nombre estándar que refleje su versión.

```
root@servidor# cp arch/x86/boot/bzImage /boot/vmlinuz-2.6.15
root@servidor#
```

f. Creación del ramdisk de módulos

Hay que dejar a disposición del kernel un ramdisk que contenga el conjunto de módulos compilados para la nueva versión. Este ramdisk necesita un archivo de imagen, que puede construirse con dos comandos distintos en función de la generación del sistema usado. El comandotradicional es **mkinitrd**. Éste tiende a desaparecer en beneficio del comando **mkinitramfs**.

Creación de un ramdisk con el comando mkinitrd

```
mkinitrd nombre_imagen versión
```

Creación de un ramdisk con el comando mkinitramfs

```
mkinitramfs -o nombre_imagen versión
```

Donde *nombre_imagen* representa el nombre del archivo de imagen de ramdisk que se desea crear y *versión* el número de versión del kernel. Este número se corresponde de hecho con el directorio de módulos albergado en /lib/modules.

Ejemplo de creación de un ramdisk

```
root@servidor:/boot$ mkinitrd /boot/initrd-2.6.28.img 2.6.28
root@servidor:/boot$ file initrd-2.6.28.img
initrd.img-2.6.32-24-generic: gzip compressed data, from Unix, last modified: Fri
Aug 20 07:54:31 2010
root@servidor:/boot$
```

El archivo ramdisk es de hecho un archivo cpio comprimido en formato gzip. Los comandos de creación de ramdisk generan directamente sus archivos en este formato.

Un sistema reciente sólo debería ofrecer el comando mkinitramfs, sin embargo, si mkinitrd también está disponible, no debería usarse. mkinitrd se basa en devfs y no en udev, y no soporta los discos sata.

g. Configuración del gestor de arranque

No basta con tener compilado el kernel y haberlo puesto en el sitio adecuado, todavía falta por configurar el gestor de arranque para que sea capaz de cargar este kernel. Por consiguiente, hay que añadir su entrada al gestor de arranque. Atención, no hay que quitar nada de lo que ya haya en la configuración del gestor de arranque: no se toca lo que ya funciona. Basta con añadir una entrada en el archivo de configuración del gestor basándose, si fuera necesario, en las entradas ya existentes.

Añadir una entrada en el gestor de arranque

La conservación de entradas existentes siempre permitirá poder arrancar usando una configuración estable.

```
# kernel operativo original
title           Debian GNU/Linux, kernel 2.6.26-2-686
root            (hd0,0)
kernel          /boot/vmlinuz-2.6.26-2-686 root=/dev/hda1 ro quiet
initrd          /boot/initrd.img-2.6.26-2-686

# kernel añadido para probar
title           PRUEBA - módulos estáticos
root            (hd0,0)
kernel          /boot/vmlinuz-2.6.20 root=/dev/hda1 ro quiet
initrd          /boot/initrd.img-2.6.20
```

Parche del kernel

1. Adición de parches

Para beneficiarse de un kernel reciente se puede descargar las fuentes completas del núcleo, compilarlas e instalarlas como un kernel nuevo. Un método alternativo consiste en utilizar las fuentes del kernel antiguo y parchearlas antes de recompilarlas.

Los parches se descargan del sitio web http://www.kernel.org y se añaden a las fuentes del kernel. La aplicación de un parche se hace generalmente con el comando **patch** y puede hacerse específicamente con un script liberado con el kernel que se llama **patch-kernel**. El script **patch-kernel** se encuentra en el directorio **scripts** de las fuentes del kernel, mientras que el comando**patch** viene con la distribución Linux.

Aplicación de un parche a las fuentes

```
patch -pn < archivo_parche
```

Aplicación de parches: opciones y parámetros	
-p*n*	Depende del diseño del archivo de parches. Sube *n* niveles jerárquicos en las rutas de los archivos escritos.
archivo_parche	El archivo que contiene los parches que se aplicarán.

Un archivo de parches es en realidad el resultado de un comando **diff** aplicado a dos estructuras de directorios de fuentes distintas. Por tanto, el archivo resultante contendrá una referencia a cada uno de los archivos de la estructura de directorios que deben modificarse. Si el nivel jerárquico de los archivos descritos no se corresponde con el de las fuentes que se modificarán, el parámetro -p permite desplazar esta jerarquía.

Ejemplo de aplicación de un parche

Los archivos de parche son extremadamente sensibles a la conformidad de las fuentes a las que se aplican. Sólo se obtendrá un resultado positivo si se aplica el parche adecuado a las fuentes adecuadas.

```
[root@beta linux-2.6.34]# patch -p1 < patch-2.6.34.4
patching file Documentation/.gitignore
patching file Documentation/hwmon/ltc4245
patching file Documentation/kernel-parameters.txt
patching file Makefile
patching file arch/arm/Kconfig
patching file arch/arm/common/sa1111.c
patching file arch/arm/include/asm/atomic.h
patching file arch/arm/include/asm/tlbflush.h
patching file arch/arm/kernel/kprobes-decode.c
patching file arch/arm/kernel/perf_event.c
patching file arch/arm/mach-mx2/devices.c
patching file arch/arm/mach-omap2/board-rx51-peripherals.c
patching file arch/arm/mach-pxa/cm-x300.c
patching file arch/arm/mach-realview/Kconfig
patching file arch/arm/mach-realview/include/mach/barriers.h
patching file arch/arm/mm/cache-v7.S
patching file arch/arm/mm/copypage-feroceon.c
patching file arch/arm/mm/copypage-v4wb.c
patching file arch/arm/mm/copypage-v4wt.c
patching file arch/arm/mm/copypage-xsc3.c
(...)
[root@beta linux-2.6.34]#
```

2. Retirada de parches

La retirada de un parche aplicado se realiza con el mismo comando y la misma sintaxis, a la que se añade el conmutador -R.

Aplicación de un parche a las fuentes

```
patch -pn -R < archivo_parche
```

Aplicación de parches: opciones y parámetros	
-p*n*	Depende del diseño del archivo de parches. Sube *n* niveles jerárquicos en las rutas de los archivos escritos.
-R	Retira el parche en vez de aplicarlo.
archivo_parche	El archivo que contiene los parches que se aplicarán.

Ejemplo de retirada de un parche

```
[root@beta linux-2.6.34]# patch -p1 -R < patch-2.6.34.4
patching file Documentation/.gitignore
patching file Documentation/hwmon/ltc4245
patching file Documentation/kernel-parameters.txt
patching file Makefile
patching file arch/arm/Kconfig
patching file arch/arm/common/sa1111.c
patching file arch/arm/include/asm/atomic.h
(...)
```

Comprobación de los conocimientos adquiridos: preguntas/respuestas

Ponga a prueba sus conocimientos respondiendo a las preguntas siguientes. Estas preguntas no siempre esperan una respuesta cerrada. Las preguntas planteadas en la certificación, a pesar de que abordan los mismos temas, en su mayoría serán planteadas en forma de preguntas de opción múltiple o que requieran una respuesta corta, en palabras escritas en el teclado.

1. Preguntas

1. ¿Por qué razón un programa escrito en un lenguaje de programación compilado es generalmente más eficiente que un programa escrito en un lenguaje interpretado?
2. ¿Para que sirve el script configure que generalmente de publica con las fuentes de los programas open source?
3. ¿Cuál es la diferencia entre los comandos make clean y make mrproper?
4. Cuando un programa se compila de forma dinámica, ¿de qué elementos depende en su entorno de ejecución?
5. ¿Cómo sabe el comando ldconfig qué directorios debe analizar para inventariar las librerías de un sistema?
6. En el contexto de una compilación de kernel, ¿qué objetivo del comando make permite basarse en un archivo .config resultante de una compilación anterior?
7. ¿Por qué motivo no debería instalarse un kernel con versión 2.5.8?
8. Ante la ausencia de una configuración particular, ¿en qué circunstancia un módulo de kernel presente en forma de archivo en el sistema no se carga en el arranque?
9. ¿Cuál es la naturaleza de un archivo de carga de ramdisk utilizado durante la carga del kernel para la detección de periféricos?
10. ¿Por qué mkinitrd ha desaparecido en beneficio del comando mkinitramfs?

2. Respuestas

1. ¿Por qué razón un programa escrito en un lenguaje de programación compilado es generalmente más eficiente que un programa escrito en un lenguaje interpretado?

   La ejecución de un programa interpretado requiere el uso de otro programa llamado intérprete, que por cada acción descrita en el código del programa deberá traducirla en una multitud de instrucciones de procesador. En el caso de un programa compilado, el código binario del programa compilado contiene directamente instrucciones inteligibles por el procesador. El tratamiento es, por tanto, mucho más rápido. Como desventaja, el código compilado está íntimamente relacionado con el juego de instrucciones de un procesador y, por consiguiente, es menos portable.

2. ¿Para que sirve el script configure que generalmente de publica con las fuentes de los programas open source?

   Para comprobar la validez del entorno de compilación y para generar un archivo de salida Makefile que utilizará el compilador. Puede ocurrir que las fuentes se publiquen sin el script configure. Generalmente sucede cuando el desarrollador no desea que se personalice su programa antes de la compilación.

3. ¿Cuál es la diferencia entre los comandos make clean y make mrproper?

   El comando make clean limpia el directorio de las fuentes de todos los elementos resultantes de la compilación. Se puede realizar una nueva compilación sobre las mismas bases. El comando make mrproper borra cualquier otro elemento que no pertenezca a las fuentes, incluso los elementos de configuración. Deja una situación similar a si se hubiera borrado todo lo realizado hasta justo después de haber extraído las fuentes del archivo tar. Cabe destacar que estas opciones no siempre están disponibles (especialmente mrproper).

4. Cuando un programa se compila de forma dinámica, ¿de qué elementos depende en su entorno de ejecución?

   De las librerías compartidas. Es muy extraño hoy en día encontrar ejecutables compilados con librerías estáticas. De este modo se optimiza el espacio en disco, pero el ejecutable se vuelve más dependiente de su

entorno.

5 ¿Cómo sabe el comando ldconfig qué directorios debe analizar para inventariar las librerías de un sistema?

Consultando el archivo de configuración /etc/ld.so.conf. Este archivo contiene la lista de directorios que se deben analizar para encontrar los archivos de librerías.

6 En el contexto de una compilación de kernel, ¿qué objetivo del comando make permite basarse en un archivo .config resultante de una compilación anterior?

Es el objetivo oldconfig. Sólo los nuevos elementos a los que no se hace referencia en el antiguo archivo .config serán objeto de pregunta al usuario.

7 ¿Por qué motivo no debería instalarse un kernel con versión 2.5.8?

Porque la numeración impar de la segunda cifra indica que se trata de un kernel en desarrollo. Las versiones de producción pasan directamente de la versión 2.4 a la versión 2.6.

8 Ante la ausencia de una configuración particular, ¿en qué circunstancia un módulo de kernel presente en forma de archivo en el sistema no se carga en el arranque?

Si este módulo no es necesario. Ya sea porque gestiona un hardware que no está en el sistema, o bien porque no se invoca por ninguna función lógica.

9 ¿Cuál es la naturaleza de un archivo de carga de ramdisk utilizado durante la carga del kernel para la detección de periféricos?

Se trata de un archivo cpio comprimido en formato gzip. Ante la ausencia de extensión estándar en un archivo comprimido, hay que recurrir al comando file para averiguar su tipo. Atención, si desea ver el contenido, primero hay que renombrarlo con un nombre que lleve la extensión adecuada (gz), y después extraerlo con el comando cpio.

10 ¿Por qué mkinitrd ha desaparecido en beneficio del comando mkinitramfs?

Porque el comando initrd no se basa en la gestión de hardware moderno udev, sino que usa devfs y, además, no es capaz de gestionar discos duros sata, lo cual es un problema en los sistemas recientes.

Trabajos prácticos

1. Compilación de una aplicación

Como lamenta la ausencia de un cliente RDP en el servidor beta, decide descargar el código fuente del cliente rdesktop y compilarlo.

a. Descarga de las fuentes

En el servidor beta, vaya al sitio www.rdesktop.org y descargue las fuentes de la última versión disponible del software (sección Downloads, la versión más reciente del programa).

Descomprima el archivo descargado en un directorio de su elección.

b. Compilación de las fuentes

Comandos útiles

- configure
- make
- tar

Operaciones

1. En el directorio de las fuentes, ejecutar el script de configuración de la compilación.
2. Compilar los fuentes.

Resumen de los comandos y resultado por pantalla

Extracción y posicionamiento en el directorio de las fuentes:

```
[root@beta rdp]# ls
rdesktop-1.6.0.tar.gz
[root@beta rdp]# tar xzf rdesktop-1.6.0.tar.gz
[root@beta rdp]# ls
rdesktop-1.6.0  rdesktop-1.6.0.tar.gz
[root@beta rdp]# cd rdesktop-1.6.0
[root@beta rdesktop-1.6.0]#
```

Configuración de la compilación:

```
[root@beta rdesktop-1.6.0]# ./configure
(...)
checking build system type... i686-redhat-linux-gnu
checking host system type... i686-redhat-linux-gnu
configure: creating ./config.status
config.status: creating Makefile
[root@beta rdesktop-1.6.0]#
```

Compilación:

c.

```
[root@beta rdesktop-1.6.0]# make
gcc -g -O2 -Wall -I/usr/include  -I/usr/include/alsa
-DPACKAGE_NAME=\"rdesktop\" -DPACKAGE_TARNAME=\"rdesktop\"
-DPACKAGE_VERSION=\"1.6.0\" -DPACKAGE_STRING=\"rdesktop\ 1.6.0\"
-DPACKAGE_BUGREPORT=\"\" -DSTDC_HEADERS=1 -DHAVE_SYS_TYPES_H=1
-DHAVE_SYS_STAT_H=1 -DHAVE_STDLIB_H=1 -DHAVE_STRING_H=1
```

```
-DHAVE_MEMORY_H=1 -DHAVE_STRINGS_H=1 -DHAVE_INTTYPES_H=1
-DHAVE_STDINT_H=1 -DHAVE_UNISTD_H=1 -DL_ENDIAN=1 -DHAVE_SYS_SELECT_H=1
-DHAVE_LOCALE_H=1 -DHAVE_LANGINFO_H=1 -Dssldir=\"/usr\"
-DEGD_SOCKET=\"/var/run/egd-pool\" -DWITH_RDPSND=1 -DRDPSND_OSS=1
-DRDPSND_ALSA=1 -DHAVE_DIRENT_H=1 -DHAVE_DIRFD=1 -DHAVE_DECL_DIRFD=1
-DHAVE_ICONV_H=1 -DHAVE_ICONV=1 -DICONV_CONST= -DHAVE_SYS_VFS_H=1
-DHAVE_SYS_STATVFS_H=1 -DHAVE_SYS_STATFS_H=1 -DHAVE_SYS_PARAM_H=1
-DHAVE_SYS_MOUNT_H=1 -DSTAT_STATVFS=1 -DHAVE_STRUCT_STATVFS_F_NAMEMAX=1
-DHAVE_STRUCT_STATFS_F_NAMELEN=1 -D_FILE_OFFSET_BITS=64
-DHAVE_MNTENT_H=1 -DHAVE_SETMNTENT=1
-DKEYMAP_PATH=\"/usr/local/share/rdesktop/keymaps/\" -o rdesktop.o -c rdesktop.c
(...)
[root@beta rdesktop-1.6.0]#
```

Instalación de los binarios

Comandos útiles

- make

Operaciones

1. Instalar los elementos compilados en el sistema.

Resumen de los comandos y resultado por pantalla

Instalación de los elementos compilados:

```
[root@beta rdesktop-1.6.0]# make install
mkdir -p /usr/local/bin
/usr/bin/install -c rdesktop /usr/local/bin
strip /usr/local/bin/rdesktop
chmod 755 /usr/local/bin/rdesktop
mkdir -p /usr/local/share/rdesktop/keymaps/
cp keymaps/?? keymaps/??-?? /usr/local/share/rdesktop/keymaps/
cp keymaps/common /usr/local/share/rdesktop/keymaps/
cp keymaps/modifiers /usr/local/share/rdesktop/keymaps/
chmod 644 /usr/local/share/rdesktop/keymaps//*
mkdir -p /usr/local/share/man/man1
cp doc/rdesktop.1 /usr/local/share/man/man1
chmod 644 /usr/local/share/man/man1/rdesktop.1
[root@beta rdesktop-1.6.0]#
```

Observe que los elementos instalados no sólo se centran en elementos binarios compilados, sino que también hay otros archivos como los de ayuda o de configuración.

d. Limpieza de fuentes

Comandos útiles

- make

Operaciones

1. Comprobar que existe el archivo ejecutable **rdesktop** en el directorio raíz de las fuentes.
2. En el directorio de las fuentes eliminar los resultados de la compilación, conservando los elementos de configuración para que puedan ser reaprovechados en una próxima compilación.
3. Comprobar que el archivo ejecutable **rdesktop** ha sido eliminado correctamente.

4. Comprobar que el archivo de salida **Makefile** se conserva en el directorio raíz de las fuentes.

Resumen de los comandos y resultado por pantalla

2.

```
[root@beta rdesktop-1.6.0]# ls -l rdesktop
-rwxr-xr-x 1 root root 615042 ago 13 21:03 rdesktop
[root@beta rdesktop-1.6.0]# make clean
rm -f *.o *~ vnc/*.o vnc/*~ rdesktop rdp2vnc
[root@beta rdesktop-1.6.0]# ls -l rdesktop
ls: no se puede acceder a rdesktop: No existe el fichero o el directorio
[root@beta rdesktop-1.6.0]# ls -l Makefile
-rw-r--r-- 1 root root 5823 ago 13 21:03 Makefile
[root@beta rdesktop-1.6.0]#
```

Compilación e instalación de un módulo de kernel

Para conseguir el uso específico de una tarjeta de red, uno de sus desarrolladores modifica el código fuente del controlador de tarjeta de red tg3. Tiene que compilar el código fuente del controlador para obtener e instalar un módulo de kernel.

a. Obtención de las fuentes

Descargue del sitio web de Ediciones ENI el archivo **linux-3.110g.tar.gz** y proceda a su extracción.

b. Eliminación del módulo existente

Comandos útiles

- lsmod
- rm
- rmmod

Operaciones

1. Comprobar que el módulo **tg3** no está actualmente cargado en memoria. Si éste fuera el caso, hay que desactivarlo con el comando apropiado.
2. Borrar el archivo **/lib/modules/2.6.x/kernel/drivers/net/tg3.ko**.

Resumen de los comandos y resultado por pantalla

c.

```
[root@beta ~] lsmod | grep tg3
[root@beta ~]
[root@beta ~] rm /lib/modules/`uname -r`/kernel/drivers/net/tg3.ko
rm: ¿borrar el fichero regular «/lib/modules/2.6.32-220.el6.i686/kernel/drivers/net/tg3.ko»?
(s/n) s
[root@beta ~]
```

Compilación de fuentes

Comandos útiles

- make

Operaciones

1. Moverse al directorio extraído y comprobar la ausencia del archivo **configure**. Los

desarolladores no han deseado que pueda interactuar con la compilación y han dejado ya escritos los archivos **Makefile** necesarios para la compilación.

2. Compilar los fuentes del controlador.

Resumen de los comandos y resultado por pantalla

Ausencia del script de configuración y archivo Makefile ya existente:

```
[root@beta tg3-3.110g]# ls -l
total 1024
-rw-r--r-- 1 2397 305 350928 abr 13 23:19 ChangeLog
-rw-r--r-- 1 2397 305  15153 ene  9  2009 LICENSE
-rw-r--r-- 1 2397 305   3870 may 13 01:16 Makefile
-rwxr--r-- 1 2397 305   6584 abr 13 18:56 makeflags.sh
-rw-r--r-- 1 2397 305  10921 jun  8 19:58 README.TXT
-rw-r--r-- 1 2397 305   3445 feb  5  2010 tg3.4
-rw-r--r-- 1 2397 305 424808 jun  9 00:45 tg3.c
-rw-r--r-- 1 2397 305   2253 mar 31 22:20 tg3_compat2.h
-rw-r--r-- 1 2397 305  35711 jun  4 20:08 tg3_compat.h
-rw-r--r-- 1 2397 305  43934 mar 31 22:26 tg3_firmware.h
-rw-r--r-- 1 2397 305 114378 jun  4 20:01 tg3.h
-rw-r--r-- 1 2397 305   4286 jun  4 01:45 tg3_vmware.c
-rw-r--r-- 1 2397 305   1354 jun  4 01:57 tg3_vmware.h
```

Compilación:

```
[root@beta tg3-3.110g]# make
sh makeflags.sh /lib/modules/2.6.18-194.el5/source > tg3_flags.h
make -C /lib/modules/2.6.18-194.el5/build SUBDIRS=/root/Desktop/red/tg3-3.110g modules
make[1]: se ingresa al directorio `/usr/src/kernels/2.6.18-194.el5-i686'
  CC [M]  /root/Desktop/red/tg3-3.110g/tg3.o
  Building modules, stage 2.
  MODPOST
  CC      /root/Desktop/red/tg3-3.110g/tg3.mod.o
  LD [M]  /root/Desktop/red/tg3-3.110g/tg3.ko
make[1]: se sale del directorio `/usr/src/kernels/2.6.18-194.el5-i686'
[root@beta tg3-3.110g]#
```

d. Carga del módulo de kernel e instalación del módulo

Comandos útiles

- insmod
- ls
- lsmod

Operaciones

1. Comprobar la presencia del archivo **tg3.ko**. Éste es el módulo del kernel que se acaba de compilar.
2. Cargar este módulo en memoria con el comando apropiado.
3. Instalar este módulo en su ubicación adecuada (prevista por el desarrollador).

Resumen de los comandos y resultado por pantalla

Carga del módulo en memoria:

```
[root@beta tg3-3.110g]# ls -l tg3.ko
-rw-r--r-- 1 root root 630546 ago 13 11:34 tg3.ko
[root@beta tg3-3.110g]# insmod tg3.ko
[root@beta tg3-3.110g]# lsmod | grep tg3
tg3                   125832  0
[root@beta tg3-3.110g]#
```

Instalación del módulo:

3.

```
[root@beta tg3-3.110g]# make install
make -C /lib/modules/2.6.18-194.el5/build SUBDIRS=/root/Desktop/red/tg3-3.110g modules
make[1]: se ingresa al directorio `/usr/src/kernels/2.6.18-194.el5-i686'
  Building modules, stage 2.
  MODPOST
make[1]: se sale del directorio `/usr/src/kernels/2.6.18-194.el5-i686'
gzip -c tg3.4 > tg3.4.gz
mkdir -p //lib/modules/2.6.18-194.el5/updates;
install -m 444 tg3.ko //lib/modules/2.6.18-194.el5/updates;
install -m 444 tg3.4.gz /usr/share/man/man4;\

[root@beta tg3-3.110g]#
```

Parchear una aplicación

Un amigo geek le pide si puede modificar su lista de la compra de la semana. Usted cree que será una buena forma de familiarizarse con el procedimiento de aplicación de parches.

a. Obtención de las fuentes y del archivo de parche

Descargue en el sitio de Ediciones ENI el archivo **compras.tar.gz** y extráigalo. Descargue también el archivo **modif_compras**.

b. Aplicación del parche

Comandos útiles

- cp
- patch
- tar

Operaciones

1. El directorio **compras** contiene una estructura de carpetas con los días de la semana y la compra que deben realizarse cada día. Aplicar el archivo de parches directamente en el directorio **compras**.
2. Mostrar la lista de compras del lunes (archivo compras en el directorio lunes).
3. El parche ha sido diseñado desde el directorio padre de los fuentes. Por lo tanto no se puede aplicar directamente desde el directorio de fuentes. Aplique el parche retirando un nivel de directorios.
4. Visualice la lista de la compra del lunes.

Resumen de los comandos y resultado por pantalla

Copia del parche:

```
toto@ubuntu:~$ ls
compras.tar.gz  modif_compras
toto@ubuntu:~$ tar xzf compras.tar.gz
```

```
toto@ubuntu:~$ ls compras
lunes  martes  miércoles
toto@ubuntu:~$ cp modif_compras compras
toto@ubuntu:~$
```

Compras del lunes:

```
toto@ubuntu:~/compras$ cat lunes/compras
mantequilla
queso
pan
zanahorias
toto@ubuntu:~/compras$
```

Aplicación del parche (sin decalado jerárquico):

```
toto@ubuntu:~/compras$ patch < modif_compras
patching file compras
Hunk #1 FAILED at 1.
1 out of 1 hunk FAILED -- saving rejects to file compras.rej
patching file compras
Hunk #1 FAILED at 1.
1 out of 1 hunk FAILED -- saving rejects to file compras.rej
patching file compras
Hunk #1 FAILED at 1.
1 out of 1 hunk FAILED -- saving rejects to file compras.rej
toto@ubuntu:~/compras$
```

Aplicación del parche:

```
toto@ubuntu:~/compras$ patch -p1 < modif_compras
patching file lunes/compras
patching file martes/compras
patching file miércoles/compras
toto@ubuntu:~/compras$
```

Compras del lunes después de aplicar el parche:

c.

```
toto@ubuntu:~/compras$ cat lunes/compras
mantequilla
queso
pan
nabos
toto@ubuntu:~/compras$
```

Retirada del parche

Como no desea comprar nabos y teme que le inviten a cenar, decide retirar el parche.

Comandos útiles

- patch

Operaciones

1. Retirar el parche.

2. Mostrar la lista de la compra del lunes.

Resumen de los comandos y resultado por pantalla

Retirada del parche:

```
toto@ubuntu:~/compras$ patch -p1 -R < modif_compras
patching file lunes/compras
patching file martes/compras
patching file miércoles/compras
toto@ubuntu:~/compras$
```

Compras del lunes después de haber retirado el parche:

4.

```
toto@ubuntu:~/compras$ cat lunes/compras
mantequilla
queso
pan
zanahorias
toto@ubuntu:~/compras$
```

Compilación e instalación de un nuevo kernel

No sabiendo cómo pasar el rato durante el fin de semana, decide compilar un nuevo kernel para su servidor alfa. Para que se pueda continuar con la versión de producción sin problemas el próximo lunes, tiene la precaución de no borrar ningún kernel existente.

a. Instalación de utilidades de compilación

Para disponer de las herramientas de compilación en el servidor alfa, hay que introducir el comando siguiente:

```
apt-get install gcc make
```

Si fuera necesario, instale la herramienta de compresión y descompresión bzip2 tecleando :

```
apt-get install bzip2
```

b. Descarga de las fuentes de un nuevo kernel

Comandos útiles

- tar

Operaciones

1. Ir al sitio web www.kernel.org y descargar las fuentes completas del último kernel estable (enlace Latest Stable Kernel).
2. Extraerlo en un directorio de trabajo neutro.

Resumen de los comandos y resultado por pantalla

c.

```
toto@alfa:~/kernel$ ls
linux-3.4.4.tar.bz2
toto@alfa:~/kernel$ tar xjf linux-3.4.4.tar.bz2
toto@alfa:~/kernel$ ls
linux-3.4.4  linux-3.4.4.tar.bz2
toto@alfa:~/kernel$
```

Configuración y compilación del kernel

Comandos útiles

- make

Operaciones

1. Generar un archivo de configuración del kernel usando todos los valores por defecto en la compilación.
2. Comprobar la existencia del archivo **.config** en el directorio raíz de los fuentes.
3. Compilar el kernel.

Resumen de los comandos y resultado por pantalla

Generación del archivo de configuración:

A

```
toto@alfa:~/kernel$ cd linux-3.4.4
toto@alfa:~/kernel/linux-3.4.4$ make defconfig
  HOSTCC  scripts/basic/fixdep
  HOSTCC  scripts/kconfig/conf.o
  SHIPPED scripts/kconfig/zconf.hash.c
  HOSTCC  scripts/kconfig/zconf.tab.o
  HOSTLD  scripts/kconf/conf
*** Default configuration is based on 'i386_defconfig'
#
# configuration written to .config
#
toto@alfa:~/kernel/linux-3.4.4$
```

menudo, puede que la definición del objetivo defconfig no sea la más adecuada para un uso habitual. Un método alternativo consiste en utilizar el objetivo config (make config) y "aburrirse" pulsando la tecla [Enter] una y otra vez para utilizar todos los valores por defecto.

Compilación del kernel:

d.

```
toto@alfa:~/kernel/linux-3.4.4$ make
scripts/kconfig/conf --silentoldconfig Kconfig
  SYSHDR  arch/x86/syscalls/../include/generated/asm/unistd_32.h
  SYSHDR  arch/x86/syscalls/../include/generated/asm/unistd_64.h
  SYSHDR  arch/x86/syscalls/../include/generated/asm/unistd_x32.h
  SYSTBL  arch/x86/syscalls/../include/generated/asm/syscalls_32.h
  HOSTCC  arch/x86/tools/relocs
  CHK     include/linux/version.h
  UPD     include/linux/version.h

(...)

  CC      arch/x86/boot/version.o
  CC      arch/x86/boot/video-vga.o
  CC      arch/x86/boot/video-vesa.o
  CC      arch/x86/boot/video-bios.o
  LD      arch/x86/boot/setup.elf
  OBJCOPY arch/x86/boot/setup.bin
  OBJCOPY arch/x86/boot/vmlinux.bin
  HOSTCC  arch/x86/boot/tools/build
  BUILD   arch/x86/boot/bzImage
Setup is 13248 bytes (padded to 13312 bytes).
System is 4812 kB
CRC 91ad448d
Kernel: arch/x86/boot/bzImage is ready  (#1)
```

```
  Building modules, stage 2.
  MODPOST 5 modules
  CC      arch/x86/kernel/test_nx.mod.o
  LD [M]  arch/x86/kernel/test_nx.ko
  CC      drivers/hid/hid-logitech-dj.mod.o
  LD [M]  drivers/hid/hid-logitech-dj.ko
  CC      drivers/scsi/scsi_wait_scan.mod.o
  LD [M]  drivers/scsi/scsi_wait_scan.ko
  CC      net/netfilter/xt_LOG.mod.o
  LD [M]  net/netfilter/xt_LOG.ko
  CC      net/netfilter/xt_mark.mod.o
  LD [M]  net/netfilter/xt_mark.ko
toto@alfa:~/kernel/linux-3.4.4$
```

Instalación del nuevo kernel y sus módulos

Comandos útiles

- cp
- make
- su

Operaciones

1. Utilizar si fuera necesario el usuario y los privilegios de la cuenta root.
2. Copiar el archivo de kernel en su directorio estándar con el nombre **vmlinuz-*versión***. Para ello, básese en la nomenclatura utilizada por su distribución para el archivo de núcleo existente.
3. Copiar el archivo **.config** de los parámetros de compilación en el directorio **/boot**con el nombre **config-*versión***.
4. Instalar los módulos de kernel en su ubicación estándar utilizando un solo comando.

Resumen de los comandos y resultado por pantalla

Instalación del kernel:

```
toto@alfa:~/kernel/linux-3.4.4$ su
Contraseña:
root@alfa:/home/toto/kernel/linux-3.4.4# cp arch/x86/boot/bzImage /boot/vmlinuz-3.4.4
root@alfa:/home/toto/kernel/linux-3.4.4#
```

Copia del archivo de

configuración:

```
root@alfa:/boot# cp /home/toto/kernel/linux-3.4.4/.config config-3.4.4
root@alfa:/boot#
```

Instalación de módulos:

```
root@alfa:/home/toto/kernel/linux-3.4.4# make modules_install
  INSTALL arch/x86/kernel/test_nx.ko
  INSTALL drivers/hid/hid-logitech-dj.ko
  INSTALL drivers/scsi/scsi_wait_scan.ko
  INSTALL net/netfilter/xt_LOG.ko
  INSTALL net/netfilter/xt_mark.ko
  DEPMOD  3.4.4
```

e.

```
root@alfa:/home/toto/kernel/linux-3.4.4#
```

Generación del ramdisk de arranque

Comandos útiles

- file
- mkinitramfs

Operaciones

1. Posicionarse en el directorio **/boot**.
2. Generar el ramdisk correspondiente a la nueva versión de kernel con el nombre**initrd.img-*versión***.
3. Si el comando no reconoce la versión del kernel, consulte el directorio**/lib/modules**.
4. Según la versión, pueden aparecer algunas advertencias.
5. Comprobar la presencia de un nuevo archivo de imagen en el directorio **/boot** y determinar su naturaleza.

Resumen de los comandos y resultado por pantalla

Generación del archivo de imagen:

```
root@alfa:/home/toto/kernel/linux-3.4.4# cd /boot
root@alfa:/boot# mkinitramfs -o initrd.img-4.5.5 3-4-4
grep: /boot/config-3-4-4: No existe el fichero o el directorio
WARNING: missing /lib/modules/3-4-4
Device driver support needs thus be built-in linux image!
FATAL: modules must be specified in absolute paths.
"3-4-4" is a relative path
FATAL: Could not load /lib/modules/3-4-4/modules.dep: No such file or directory
(...)
root@alfa:/boot# ls /lib/modules/
2.6.32-5-686  3.4.4
root@alfa:/boot# mkinitramfs -o initrd.img-3.4.4 3.4.4
W: mdadm: /etc/mdadm/mdadm.conf defines no arrays.
root@alfa:/boot#
```

Comprobación:

f.

```
root@alfa:/boot# file ini*
initrd.img-2.6.32-5-686: gzip compresses data, from Unix, last modified: Mon Oct
1 20:44:32 2012
initrd.img-3.4.4:        gzip compresses data, from Unix, last modified: Wed Oct
3 18:28:51 2012
root@alfa:/boot#
```

Configuración del gestor de arranque GRUB 2

Comandos útiles

- update-grub

Operaciones

1. Solicitar al gestor de arranque que tenga en cuenta al nuevo núcleo.
2. Reiniciar el servidor y elegir el núcleo.
3. No se deje impresionar por los mensajes de error, ya sean bloqueantes o no. La compilación de un núcleo es una tarea laboriosa y no siempre se acaba en medio día. ¡Tiene que conservar los núcleos existentes por si aparece algún error fatal!

Resumen de los comandos y resultado por pantalla

g.

```
root@alfa:/boot# update-grub
Generating grub.cfg ...
Found linux image: /boot/vmlinuz-3.4.4
Found initrd image: /boot/initrd.img-3.4.4
Found linux image: /boot/vmlinuz-2.6.32-5-686
Found initrd image: /boot/initrd.img-2.6.32-5-686
done
root@alfa:/boot#
```

Configuración del gestor de arranque GRUB 1 (alternativa)

Si realiza la instalación del núcleo en un sistema con GRUB 1, como una Debian 5 o la distribución CentOS, siga los siguientes pasos :

Comandos y archivos útiles

- /boot/grub/menu.lst
- vi

Operaciones

1. Editar el archivo de configuración del gestor de arranque grub.
2. Aumentar el valor de timeout para tener más tiempo en el siguiente arranque.
3. Al final del archivo, se encuentra la última sección que hace referencia a un kernel ordinario (non single), duplicarla.
4. Modificar el título de forma que aparezca de algún modo que el kernel no está validado.
5. Modificar el valor del parámetro kernel para cargar el nuevo kernel.
6. Modificar el valor del parámetro initrd para cambiar al nuevo archivo de imágenes de módulos.
7. Reiniciar el servidor y elegir el nuevo kernel.
8. No hay que dejarse impresionar por los mensajes de error, ya sean bloqueantes o no. La compilación de un kernel es una tarea larga y no tiene éxito necesariamente en medio día. Además, ¡hay que recordar que se tiene que conservar los kernels existentes por si aparecen errores!

Resumen de los comandos y resultado por pantalla

Archivo /boot/grub/menu.lst modificado:

```
timeout 15
(...)
## ## End Default Options ##

title           Debian GNU/Linux, kernel 2.6.26-2-686
root            (hd0,0)
kernel          /boot/vmlinuz-2.6.26-2-686 root=/dev/hda1 ro quiet
```

```
initrd          /boot/initrd.img-2.6.26-2-686

title           Debian GNU/Linux, kernel 2.6.26-2-686 (single-user mode)
root            (hd0,0)
kernel          /boot/vmlinuz-2.6.26-2-686 root=/dev/hda1 ro single
initrd          /boot/initrd.img-2.6.26-2-686

### END DEBIAN AUTOMAGIC KERNELS LIST
title           PRUEBA - No utilizar en producción - kernel 2.6.35
root            (hd0,0)
kernel          /boot/vmlinuz-2.6.35 root=/dev/hda1 ro quiet
initrd          /boot/initrd.img-2.6.35
```

Tabla de objetivos

Para los trabajos prácticos, los casos marcados con un (1) indican la ausencia de trabajos prácticos (tema estrictamente teórico del que no se puede realizar ejercicios). Las referencias entre paréntesis indican un tratamiento conjunto del tema en el interior de otro trabajo práctico. Finalmente, los casos marcados con un (2) indican elementos que están repartidos por todo el libro y, por consiguiente, se tratan en todos los capítulos y ejercicios (la resolución de problemas -troubleshooting- se trata en el conjunto de materias).

	Capítulo	Trabajos prácticos
Exam 201: Detailed Objectives		
Topic 201: Linux Kernel		
201.1 Kernel Components	Compilación de aplicaciones y del kernel Linux - Compilación del kernel - Los componentes del kernel	(1)
201.2 Compiling a kernel	Compilación de aplicaciones y del kernel Linux - Compilación del kernel - Procedimiento de compilación y de utilización	Compilación e instalación de un nuevo kernel
201.3 Patching a kernel	Compilación de aplicaciones y del kernel Linux - Parche del kernel - Adición de parches	Parchear una aplicación
201.4 Customise, build and install a custom kernel and kernel modules	Compilación de aplicaciones y del kernel Linux - Compilación del kernel - Procedimiento de compilación y de utilización	Compilación e instalación de un módulo de kernel
201.5 Manage/Query kernel and kernel modules at runtime	Compilación de aplicaciones y del kernel Linux - Compilación del kernel - Los componentes del kernel	Compilación e instalación de un módulo de kernel
Topic 202: System Startup		
202.1 Customising system startup and boot processes	Arranque del sistema - El proceso init y los niveles de ejecución - Configuración del proceso init	Creación de un nivel de ejecución personalizado con aplicaciones específicas
202.2 System recovery	Arranque del sistema - Arranque y carga del kernel	Reinstalación de GRUB 1 después de haberse corrompido
Topic 203: Filesystem and Devices		
203.1 Operating the Linux filesystem	Administración del almacenamiento - Administración y configuración de sistemas de archivos - Montaje de sistemas de archivos	Creación y uso de un volumen lógico en el disco RAID 0
203.2 Maintaining a Linux filesystem	Administración del almacenamiento - Administración y configuración de sistemas de archivos - Administración de	Ampliación del volumen lógico

	sistemas de archivos Administración del almacenamiento - Copias de seguridad - Copias de seguridad a nivel de sistema de archivos	
203.3 Creating and configuring filesystem options	Administración del almacenamiento - Administración y configuración de sistemas de archivos - Protección de datos almacenados Administración del almacenamiento - Copias de seguridad - Duplicación y sincronización de datos	Creación y uso de un volumen lógico en el disco RAID 0
203.4 udev Device Management	Administración del almacenamiento - Administración y configuración de sistemas de archivos - Administración de discos duros	(1)
Topic 204: Advanced Storage Device Administration		
204.1 Configuring RAID	Administración del almacenamiento - RAID - Configuración de RAID	Configuración de un disco en RAID 0
204.2 Adjusting Storage Device Access	Administración del almacenamiento - Administración y configuración de sistemas de archivos - Administración de discos duros	Configuración de un disco en RAID 0
204.3 Logical Volume Manager	Administración del almacenamiento - Logical Volume Manager	Creación y uso de un volumen lógico en el disco RAID 0 - Ampliación del volumen lógico
Topic 205: Networking Configuration		
205.1 Basic networking configuration	Administración de la red local - Configuración de la red - Configuración universal de la red	Configuración de un servidor DHCP en el servidor alfa
205.2 Advanced Network Configuration and Troubleshooting	Administración de la red local - Diagnóstico de red - Herramientas de diagnóstico en la capa de red	Configuración de un servidor DHCP en el servidor alfa
205.3 Troubleshooting network issues	Administración de la red local - Diagnóstico de red - Herramientas de diagnóstico en las capas de transporte y de aplicación	Uso del servicio DHCP
205.4 Notify users on system-related issues	Correo electrónico - Recepción local de mensajes - Alternativas al correo	(1)

Topic 206: System Maintenance		
206.1 Make and install programs from source	Compilación de aplicaciones y del kernel Linux - Compilación de aplicaciones - Procedimiento de compilación GNU	Compilación de una aplicación
206.2 Backup operations	Administración del almacenamiento - Copias de seguridad - Las herramientas de archivado	(2)
Topic 207: Domain Name Server		
207.1 Basic DNS server configuration	Resolución de nombres DNS - Configuración básica del servidor - Servidor de caché	Configuración del servidor de caché
207.2 Create and maintain DNS zones	Resolución de nombres DNS - Administración de zonas DNS	Creación de zonas personalizadas directas e inversas
207.3 Securing a DNS server	Resolución de nombres DNS - Seguridad en el servicio DNS	Creación de un servidor secundario
Exam 202: Detailed Objectives		
Topic 208: Web Services		
208.1 Implementing a web server	Servidor web Apache - Configuración básica de un servidor Apache	Restricción de acceso a páginas web
208.2 Maintaining a web server	Servidor web Apache - Configuración de Apache con SSL	Autentificación local
208.3 Implementing a proxy server	Servidor web Apache - Servidor proxy - El servidor proxy squid	Autentificación mediante directorio LDAP
Topic 209: File Sharing		
209.1 SAMBA Server Configuration	Compartición de archivos - Compartición de datos con Samba	Despliegue de comparticiones Samba en el servidor alfa
209.2 NFS Server Configuration	Compartición de archivos - Compartición de datos con NFS	Despliegue de comparticiones NFS en el servidor beta
Topic 210: Network Client Management		
210.1 DHCP configuration	Administración de la red local - Configuración automática con DHCP	Configuración de un servidor DHCP en el servidor alfa Uso del servicio DHCP
210.2 PAM authentication	Autentificación de usuarios - PAM	Autentificación del puesto de trabajo mediante el directorio LDAP
210.3 LDAP client usage	Autentificación de usuarios - LDAP - Herramientas LDAP cliente	Creación y alimentación de un directorio LDAP en el servidor beta
Topic 211: E-Mail Services		

211.1 Using e-mail servers	Correo electrónico - Los MTA	Gestión de los envíos
211.2 Managing Local E-Mail Delivery	Correo electrónico - Recepción local de mensajes	Gestión de los envíos
211.3 Managing Remote E-Mail Delivery	Correo electrónico - Recepción remota de mensajes	Gestion de las recepciones
Topic 212: System Security		
212.1 Configuring a router	Protección de redes - Enrutamiento y filtrado	Configuración del router y del cortafuegos en el servidor beta
212.2 Securing FTP servers	Compartición de archivos - Compartición de archivos con FTP	Configuración de un servidor FTP en el servidor alfa
212.3 Secure shell (SSH)	Asegurar las comunicaciones - OpenSSH	Creación de un túnel SSH entre la estación de trabajo y el servidor beta
212.4 TCP Wrapper	Administración de la red local - Configuración de la red - Otros comandos y archivos de administración de la red	(1)
212.5 Security tasks	Protección de redes - Administración de un cortafuegos con iptables - Filtrado de paquetes	Configuración del router y del cortafuegos en el servidor beta
Topic 213: Troubleshooting		
213.1 Identifying boot stages and troubleshooting bootloaders	Arranque del sistema - Arranque y carga del kernel - Utilización de GRUB 1 en modo interactivo, Reinstalación de GRUB	Reinstalación de GRUB 1 después de haberse corrompido
213.2 General troubleshooting	(2)	(2)
213.3 Troubleshooting system resources	(2)	(2)
213.4 Troubleshooting environment configurations	(2)	(2)

Colección

Certificaciones

Los exámenes **LPI 201 y LPI 202** son los dos exámenes que permiten obtener la **certificación LPIC-2** "Advanced Level Linux Professional". Este programa de certificación del Linux Professional Institute está cada vez más **reconocido por las empresas de selección**, que ven en esta certificación un requisito para la contratación o el acceso a un puesto de administrador.

Los exámenes LPI 201 y 202 demuestran a los profesionales que usted domina la administración avanzada de un sistema **Linux de cualquier distribución**: califican las competencias prácticas en términos de administración de redes de pequeño o mediano tamaño (administración de servicios de red comunes, gestión de la seguridad de red y de las comunicaciones...).

Para ayudarle a preparar eficazmente esta certificación, este libro cubre **todos los objetivos oficiales de la última versión del examen** (implantada en julio de 2012) tanto desde un punto de vista teórico como práctico. Ha sido originalmente redactado por un profesional formador y consultor, certificado en Linux. De este modo, el saber pedagógico y técnico del autor conducen a una aproximación clara y visual, de un nivel técnico muy alto.

Capítulo a capítulo, podrá **validar sus conocimientos teóricos**, con la ayuda de múltiples **preguntas-respuestas (110 en total)** que ponen de relieve tanto los elementos fundamentales como las características específicas de los conceptos tratados.

Cada capítulo finaliza con unos **trabajos prácticos (32 en total)** en los que puede medir su autonomía. Estas operaciones concretas, más allá incluso de los objetivos marcados para el examen, le permitirán construir una primera experiencia significativa y adquirir verdaderas competencias técnicas en situaciones reales.

A este dominio de práctica y de conocimientos, se añade la preparación específica para la certificación: podrá acceder **gratuitamente a 1 examen en blanco en línea**, para que pueda practicar en condiciones parecidas a las de la prueba.

Después de haber sido Administrador de Sistemas y Redes, **Sébastien BOBILLIER** evoluciona durante muchos años en el mundo de la formación. Hoy en día, es Consultor Formador en Global Knowledge, y se ha convertido en un especialista de sistemas Linux, que acompaña regularmente a los candidatos a la certificación LPI. Este libro es el fruto de toda su experiencia en esta materia.

Los capítulos del libro

Descripción • Introducción • Administración del almacenamiento • Arranque del sistema • Administración de la red local • Autentificación de usuarios • Compartición de archivos • Resolución de nombres DNS • Servidor web Apache • Correo electrónico • Protección de redes • Asegurar las comunicaciones • Compilación de aplicaciones y del kernel Linux • Objetivos

En **www.ediciones-eni.com**:
➜ Archivos de configuración de los servidores resultantes de los trabajos prácticos.

ISBN: 978-2-7460-7926-7

EXÁMENES LPI 201 y LPI 202

ediciones